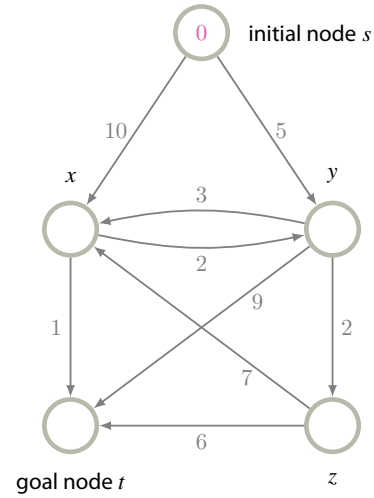
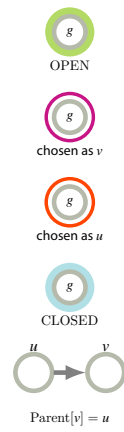


```

1 function Dijkstra(s)
2   OPEN ← new PriorityQueueg
3   g[s] ← 0
4   Insertg(OPEN, s)
5   CLOSED ← ∅
6   loop do
7     if IsEmpty(OPEN) then
8       return "failure"
9     v ← DeleteMing(OPEN)
10    CLOSED ← CLOSED ∪ {v}
11    if IsGoal(v) then
12      return Solution(v, s)
13    Expand(v)
14  procedure Expand(v)
15  foreach u ∈ Succ(v) do
16    if u ∉ OPEN ∪ CLOSED then
17      g[u] ← g[v] + c(v, u)
18      Parent[u] ← v
19      Insertg(OPEN, u)
20    else if u ∈ OPEN then
21      if g[v] + c(v, u) < g[u] then
22        g[u] ← g[v] + c(v, u)
23        Parent[u] ← v

```

1

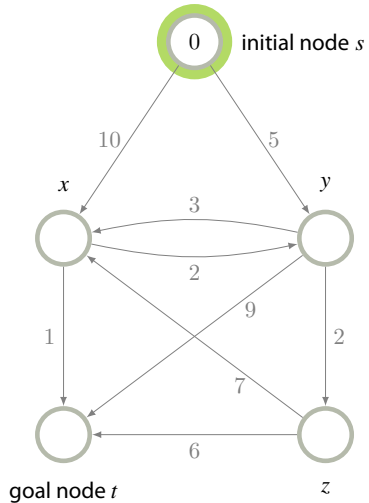


```

1 function Dijkstra(s)
2   OPEN ← new PriorityQueueg
3   g[s] ← 0
4   Insertg(OPEN, s)
5   CLOSED ← ∅
6   loop do
7     if IsEmpty(OPEN) then
8       return "failure"
9     v ← DeleteMing(OPEN)
10    CLOSED ← CLOSED ∪ {v}
11    if IsGoal(v) then
12      return Solution(v, s)
13    Expand(v)
14  procedure Expand(v)
15  foreach u ∈ Succ(v) do
16    if u ∉ OPEN ∪ CLOSED then
17      g[u] ← g[v] + c(v, u)
18      Parent[u] ← v
19      Insertg(OPEN, u)
20    else if u ∈ OPEN then
21      if g[v] + c(v, u) < g[u] then
22        g[u] ← g[v] + c(v, u)
23        Parent[u] ← v

```

2

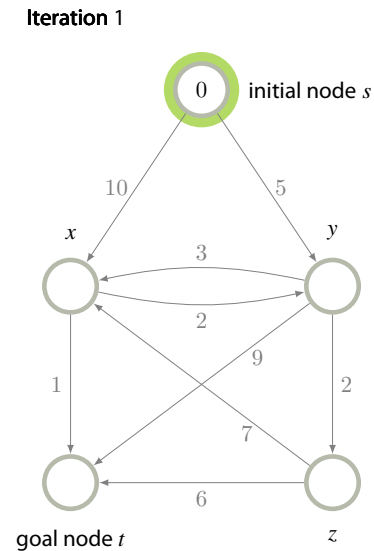
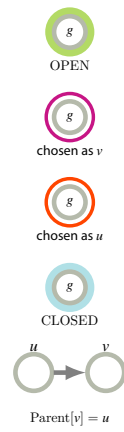


```

1 function Dijkstra(s)
2   OPEN ← new PriorityQueueg
3   g[s] ← 0
4   Insertg(OPEN, s)
5   CLOSED ← ∅
6   loop do
7     if IsEmpty(OPEN) then
8       return "failure"
9     v ← DeleteMing(OPEN)
10    CLOSED ← CLOSED ∪ {v}
11    if IsGoal(v) then
12      return Solution(v, s)
13    Expand(v)
14  procedure Expand(v)
15  foreach u ∈ Succ(v) do
16    if u ∉ OPEN ∪ CLOSED then
17      g[u] ← g[v] + c(v, u)
18      Parent[u] ← v
19      Insertg(OPEN, u)
20    else if u ∈ OPEN then
21      if g[v] + c(v, u) < g[u] then
22        g[u] ← g[v] + c(v, u)
23        Parent[u] ← v

```

3

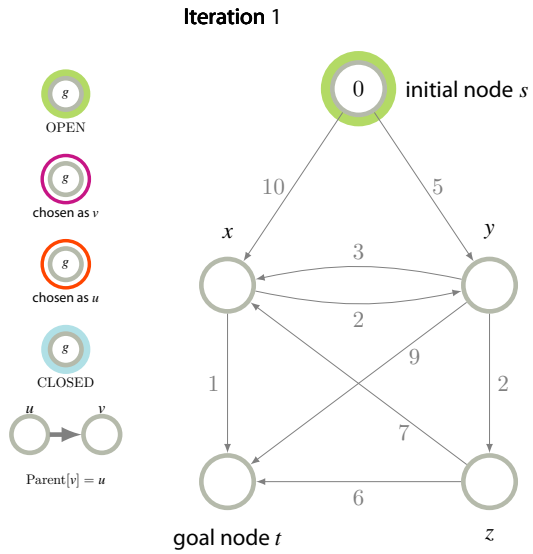


```

1 function Dijkstra(s)
2   OPEN ← new PriorityQueueg
3   g[s] ← 0
4   Insertg(OPEN, s)
5   CLOSED ← ∅
6   loop do
7     if IsEmpty(OPEN) then
8       return "failure"
9     v ← DeleteMing(OPEN)
10    CLOSED ← CLOSED ∪ {v}
11    if IsGoal(v) then
12      return Solution(v, s)
13    Expand(v)
14  procedure Expand(v)
15  foreach u ∈ Succ(v) do
16    if u ∉ OPEN ∪ CLOSED then
17      g[u] ← g[v] + c(v, u)
18      Parent[u] ← v
19      Insertg(OPEN, u)
20    else if u ∈ OPEN then
21      if g[v] + c(v, u) < g[u] then
22        g[u] ← g[v] + c(v, u)
23        Parent[u] ← v

```

4

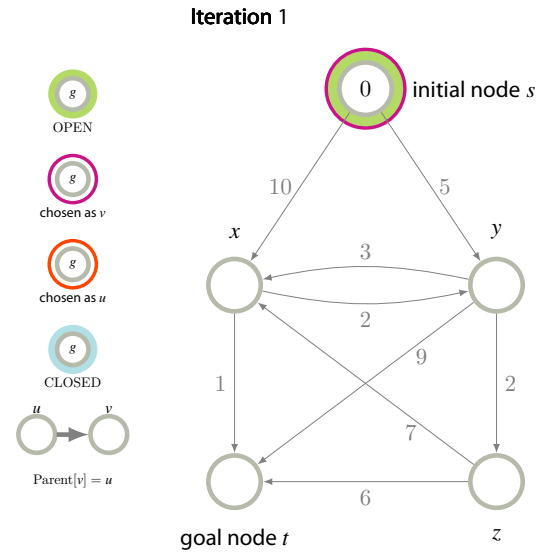


```

1 function Dijkstra(s)
2 OPEN ← new PriorityQueueg
3 g[s] ← 0
4 Insertg(OPEN, s)
5 CLOSED ← ∅
6 loop do
7   if IsEmpty(OPEN) then
8     return "failure"
9   v ← DeleteMing(OPEN)
10  CLOSED ← CLOSED ∪ {v}
11  if IsGoal(v) then
12    return Solution(v, s)
13  Expand(v)
14 procedure Expand(v)
15 foreach u ∈ Succ(v) do
16   if u ∉ OPEN ∪ CLOSED then
17     g[u] ← g[v] + c(v, u)
18     Parent[u] ← v
19     Insertg(OPEN, u)
20   else if u ∈ OPEN then
21     if g[v] + c(v, u) < g[u] then
22       g[u] ← g[v] + c(v, u)
23       Parent[u] ← v

```

5

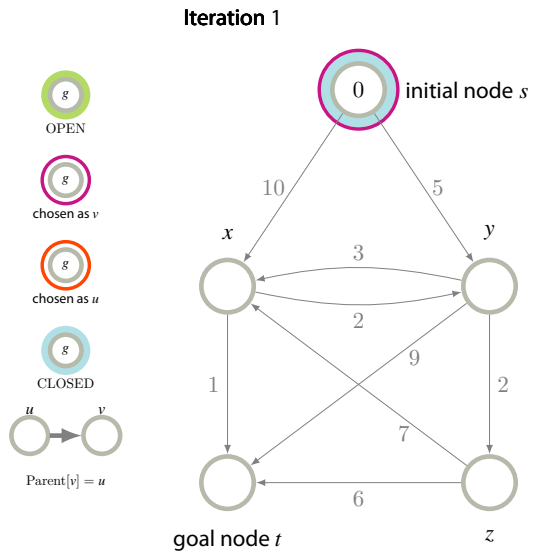


```

1 function Dijkstra(s)
2 OPEN ← new PriorityQueueg
3 g[s] ← 0
4 Insertg(OPEN, s)
5 CLOSED ← ∅
6 loop do
7   if IsEmpty(OPEN) then
8     return "failure"
9   v ← DeleteMing(OPEN)
10  CLOSED ← CLOSED ∪ {v}
11  if IsGoal(v) then
12    return Solution(v, s)
13  Expand(v)
14 procedure Expand(v)
15 foreach u ∈ Succ(v) do
16   if u ∉ OPEN ∪ CLOSED then
17     g[u] ← g[v] + c(v, u)
18     Parent[u] ← v
19     Insertg(OPEN, u)
20   else if u ∈ OPEN then
21     if g[v] + c(v, u) < g[u] then
22       g[u] ← g[v] + c(v, u)
23       Parent[u] ← v

```

6

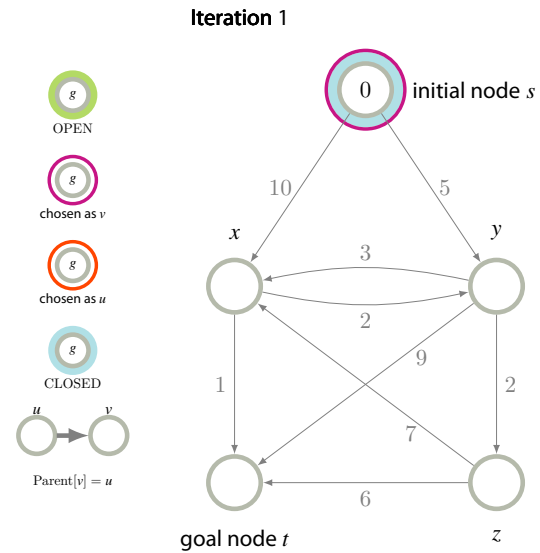


```

1 function Dijkstra(s)
2 OPEN ← new PriorityQueueg
3 g[s] ← 0
4 Insertg(OPEN, s)
5 CLOSED ← ∅
6 loop do
7   if IsEmpty(OPEN) then
8     return "failure"
9   v ← DeleteMing(OPEN)
10  CLOSED ← CLOSED ∪ {v}
11  if IsGoal(v) then
12    return Solution(v, s)
13  Expand(v)
14 procedure Expand(v)
15 foreach u ∈ Succ(v) do
16   if u ∉ OPEN ∪ CLOSED then
17     g[u] ← g[v] + c(v, u)
18     Parent[u] ← v
19     Insertg(OPEN, u)
20   else if u ∈ OPEN then
21     if g[v] + c(v, u) < g[u] then
22       g[u] ← g[v] + c(v, u)
23       Parent[u] ← v

```

7

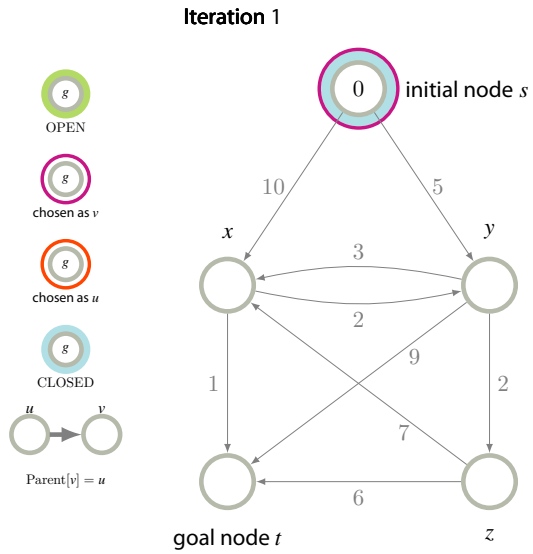


```

1 function Dijkstra(s)
2 OPEN ← new PriorityQueueg
3 g[s] ← 0
4 Insertg(OPEN, s)
5 CLOSED ← ∅
6 loop do
7   if IsEmpty(OPEN) then
8     return "failure"
9   v ← DeleteMing(OPEN)
10  CLOSED ← CLOSED ∪ {v}
11  if IsGoal(v) then
12    return Solution(v, s)
13  Expand(v)
14 procedure Expand(v)
15 foreach u ∈ Succ(v) do
16   if u ∉ OPEN ∪ CLOSED then
17     g[u] ← g[v] + c(v, u)
18     Parent[u] ← v
19     Insertg(OPEN, u)
20   else if u ∈ OPEN then
21     if g[v] + c(v, u) < g[u] then
22       g[u] ← g[v] + c(v, u)
23       Parent[u] ← v

```

8

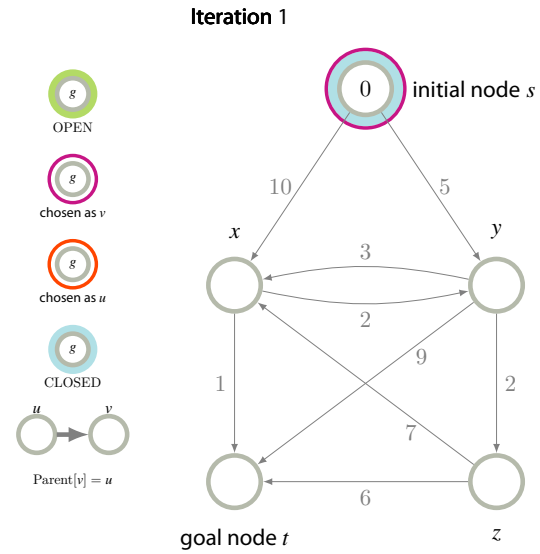


```

1 function Dijkstra(s)
2   OPEN ← new PriorityQueueg
3   g[s] ← 0
4   Insertg(OPEN, s)
5   CLOSED ← ∅
6   loop do
7     if IsEmpty(OPEN) then
8       return "failure"
9     v ← DeleteMing(OPEN)
10    CLOSED ← CLOSED ∪ {v}
11    if IsGoal(v) then
12      return Solution(v, s)
13    Expand(v)
14  procedure Expand(v)
15  foreach u ∈ Succ(v) do
16    if u ∉ OPEN ∪ CLOSED then
17      g[u] ← g[v] + c(v, u)
18      Parent[u] ← v
19      Insertg(OPEN, u)
20    else if u ∈ OPEN then
21      if g[v] + c(v, u) < g[u] then
22        g[u] ← g[v] + c(v, u)
23        Parent[u] ← v

```

9

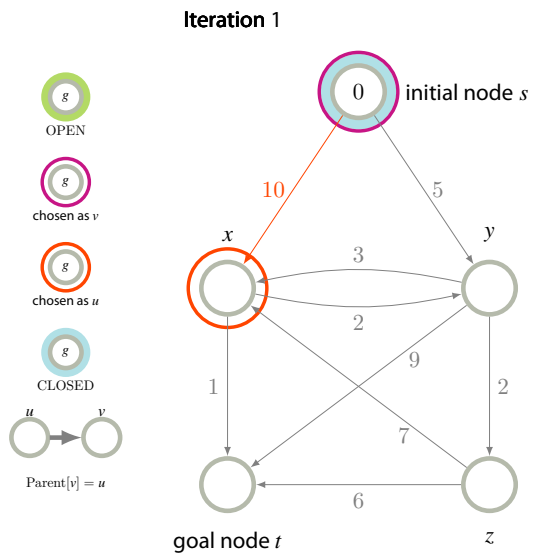


```

1 function Dijkstra(s)
2   OPEN ← new PriorityQueueg
3   g[s] ← 0
4   Insertg(OPEN, s)
5   CLOSED ← ∅
6   loop do
7     if IsEmpty(OPEN) then
8       return "failure"
9     v ← DeleteMing(OPEN)
10    CLOSED ← CLOSED ∪ {v}
11    if IsGoal(v) then
12      return Solution(v, s)
13    Expand(v)
14  procedure Expand(v)
15  foreach u ∈ Succ(v) do
16    if u ∉ OPEN ∪ CLOSED then
17      g[u] ← g[v] + c(v, u)
18      Parent[u] ← v
19      Insertg(OPEN, u)
20    else if u ∈ OPEN then
21      if g[v] + c(v, u) < g[u] then
22        g[u] ← g[v] + c(v, u)
23        Parent[u] ← v

```

10

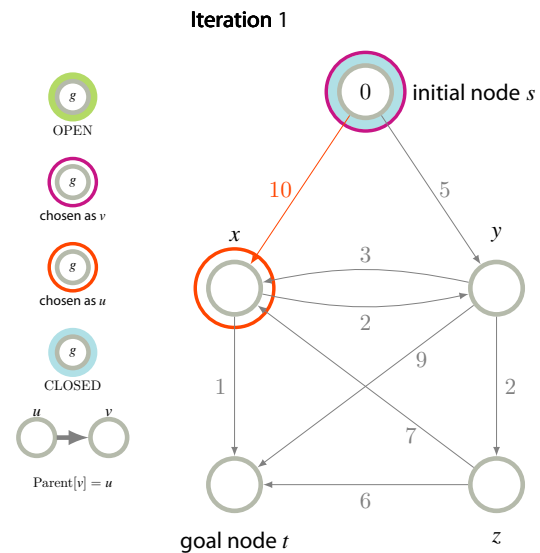


```

1 function Dijkstra(s)
2   OPEN ← new PriorityQueueg
3   g[s] ← 0
4   Insertg(OPEN, s)
5   CLOSED ← ∅
6   loop do
7     if IsEmpty(OPEN) then
8       return "failure"
9     v ← DeleteMing(OPEN)
10    CLOSED ← CLOSED ∪ {v}
11    if IsGoal(v) then
12      return Solution(v, s)
13    Expand(v)
14  procedure Expand(v)
15  foreach u ∈ Succ(v) do
16    if u ∉ OPEN ∪ CLOSED then
17      g[u] ← g[v] + c(v, u)
18      Parent[u] ← v
19      Insertg(OPEN, u)
20    else if u ∈ OPEN then
21      if g[v] + c(v, u) < g[u] then
22        g[u] ← g[v] + c(v, u)
23        Parent[u] ← v

```

11

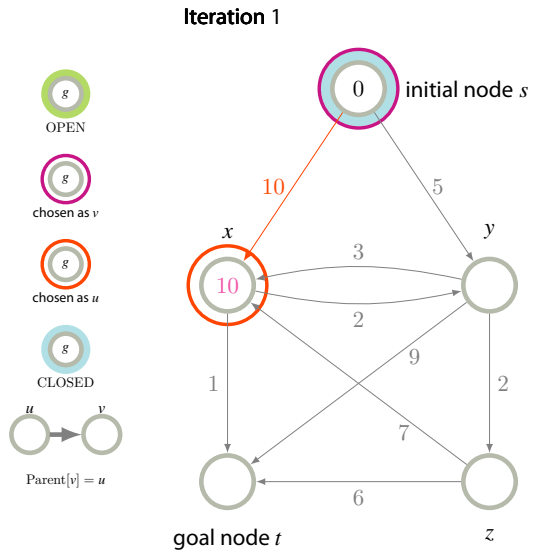


```

1 function Dijkstra(s)
2   OPEN ← new PriorityQueueg
3   g[s] ← 0
4   Insertg(OPEN, s)
5   CLOSED ← ∅
6   loop do
7     if IsEmpty(OPEN) then
8       return "failure"
9     v ← DeleteMing(OPEN)
10    CLOSED ← CLOSED ∪ {v}
11    if IsGoal(v) then
12      return Solution(v, s)
13    Expand(v)
14  procedure Expand(v)
15  foreach u ∈ Succ(v) do
16    if u ∉ OPEN ∪ CLOSED then
17      g[u] ← g[v] + c(v, u)
18      Parent[u] ← v
19      Insertg(OPEN, u)
20    else if u ∈ OPEN then
21      if g[v] + c(v, u) < g[u] then
22        g[u] ← g[v] + c(v, u)
23        Parent[u] ← v

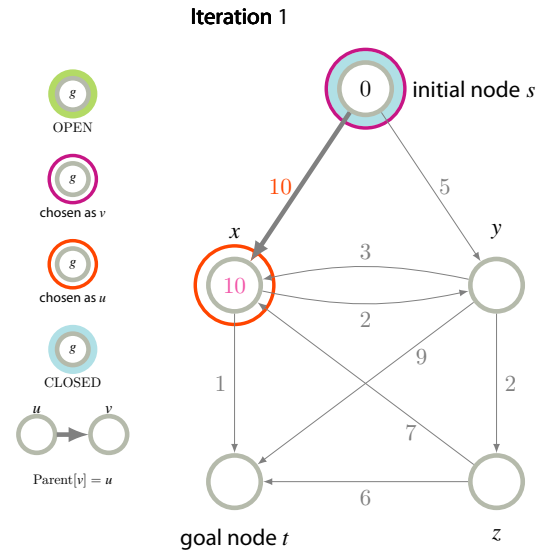
```

12



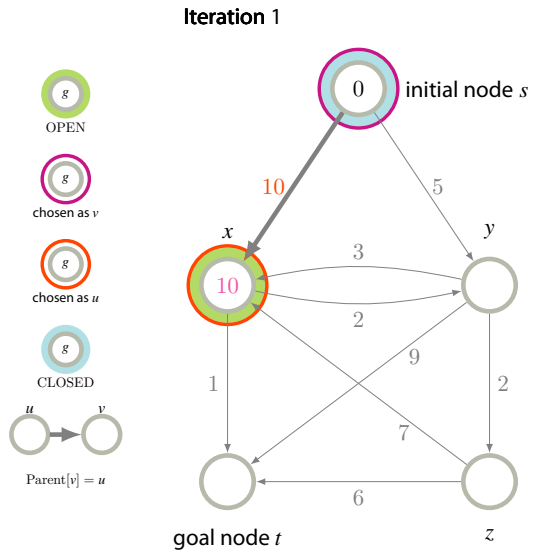
```

1 function Dijkstra( $s$ )
2 OPEN  $\leftarrow$  new PriorityQueue $_g$ 
3  $g[s] \leftarrow 0$ 
4 Insert $_g$ (OPEN,  $s$ )
5 CLOSED  $\leftarrow \emptyset$ 
6 loop do
7   if IsEmpty(OPEN) then
8     return "failure"
9    $v \leftarrow$  DeleteMin $_g$ (OPEN)
10  CLOSED  $\leftarrow$  CLOSED  $\cup \{v\}$ 
11  if IsGoal( $v$ ) then
12    return Solution( $v, s$ )
13  Expand( $v$ )
14 procedure Expand( $v$ )
15 foreach  $u \in$  Succ( $v$ ) do
16   if  $u \notin$  OPEN  $\cup$  CLOSED then
17      $g[u] \leftarrow g[v] + c(v, u)$ 
18     Parent[ $u$ ]  $\leftarrow v$ 
19     Insert $_g$ (OPEN,  $u$ )
20   else if  $u \in$  OPEN then
21     if  $g[v] + c(v, u) < g[u]$  then
22        $g[u] \leftarrow g[v] + c(v, u)$ 
23       Parent[ $u$ ]  $\leftarrow v$ 
  
```



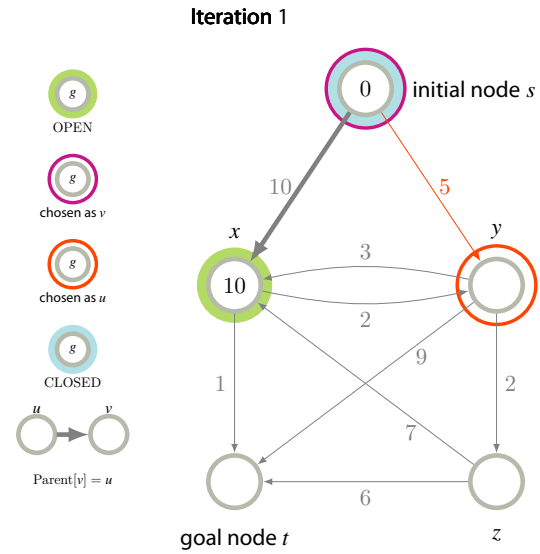
```

1 function Dijkstra( $s$ )
2 OPEN  $\leftarrow$  new PriorityQueue $_g$ 
3  $g[s] \leftarrow 0$ 
4 Insert $_g$ (OPEN,  $s$ )
5 CLOSED  $\leftarrow \emptyset$ 
6 loop do
7   if IsEmpty(OPEN) then
8     return "failure"
9    $v \leftarrow$  DeleteMin $_g$ (OPEN)
10  CLOSED  $\leftarrow$  CLOSED  $\cup \{v\}$ 
11  if IsGoal( $v$ ) then
12    return Solution( $v, s$ )
13  Expand( $v$ )
14 procedure Expand( $v$ )
15 foreach  $u \in$  Succ( $v$ ) do
16   if  $u \notin$  OPEN  $\cup$  CLOSED then
17      $g[u] \leftarrow g[v] + c(v, u)$ 
18     Parent[ $u$ ]  $\leftarrow v$ 
19     Insert $_g$ (OPEN,  $u$ )
20   else if  $u \in$  OPEN then
21     if  $g[v] + c(v, u) < g[u]$  then
22        $g[u] \leftarrow g[v] + c(v, u)$ 
23       Parent[ $u$ ]  $\leftarrow v$ 
  
```



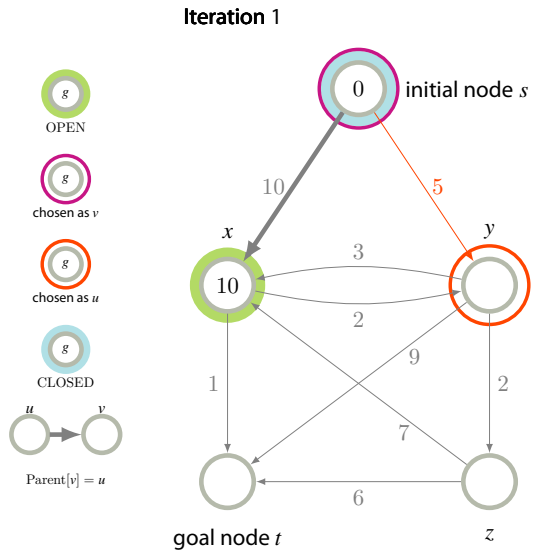
```

1 function Dijkstra( $s$ )
2 OPEN  $\leftarrow$  new PriorityQueue $_g$ 
3  $g[s] \leftarrow 0$ 
4 Insert $_g$ (OPEN,  $s$ )
5 CLOSED  $\leftarrow \emptyset$ 
6 loop do
7   if IsEmpty(OPEN) then
8     return "failure"
9    $v \leftarrow$  DeleteMin $_g$ (OPEN)
10  CLOSED  $\leftarrow$  CLOSED  $\cup \{v\}$ 
11  if IsGoal( $v$ ) then
12    return Solution( $v, s$ )
13  Expand( $v$ )
14 procedure Expand( $v$ )
15 foreach  $u \in$  Succ( $v$ ) do
16   if  $u \notin$  OPEN  $\cup$  CLOSED then
17      $g[u] \leftarrow g[v] + c(v, u)$ 
18     Parent[ $u$ ]  $\leftarrow v$ 
19     Insert $_g$ (OPEN,  $u$ )
20   else if  $u \in$  OPEN then
21     if  $g[v] + c(v, u) < g[u]$  then
22        $g[u] \leftarrow g[v] + c(v, u)$ 
23       Parent[ $u$ ]  $\leftarrow v$ 
  
```



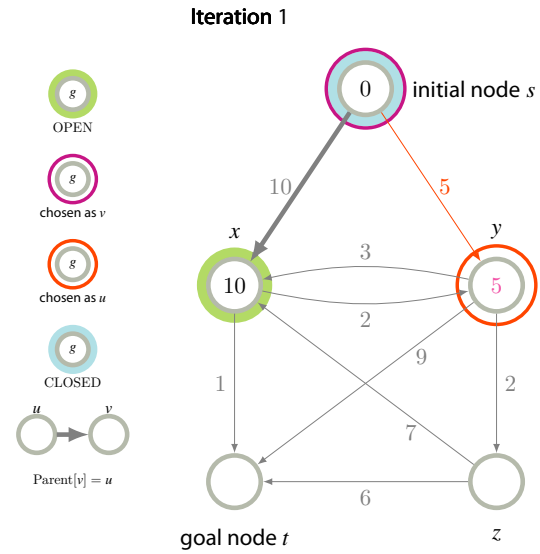
```

1 function Dijkstra( $s$ )
2 OPEN  $\leftarrow$  new PriorityQueue $_g$ 
3  $g[s] \leftarrow 0$ 
4 Insert $_g$ (OPEN,  $s$ )
5 CLOSED  $\leftarrow \emptyset$ 
6 loop do
7   if IsEmpty(OPEN) then
8     return "failure"
9    $v \leftarrow$  DeleteMin $_g$ (OPEN)
10  CLOSED  $\leftarrow$  CLOSED  $\cup \{v\}$ 
11  if IsGoal( $v$ ) then
12    return Solution( $v, s$ )
13  Expand( $v$ )
14 procedure Expand( $v$ )
15 foreach  $u \in$  Succ( $v$ ) do
16   if  $u \notin$  OPEN  $\cup$  CLOSED then
17      $g[u] \leftarrow g[v] + c(v, u)$ 
18     Parent[ $u$ ]  $\leftarrow v$ 
19     Insert $_g$ (OPEN,  $u$ )
20   else if  $u \in$  OPEN then
21     if  $g[v] + c(v, u) < g[u]$  then
22        $g[u] \leftarrow g[v] + c(v, u)$ 
23       Parent[ $u$ ]  $\leftarrow v$ 
  
```



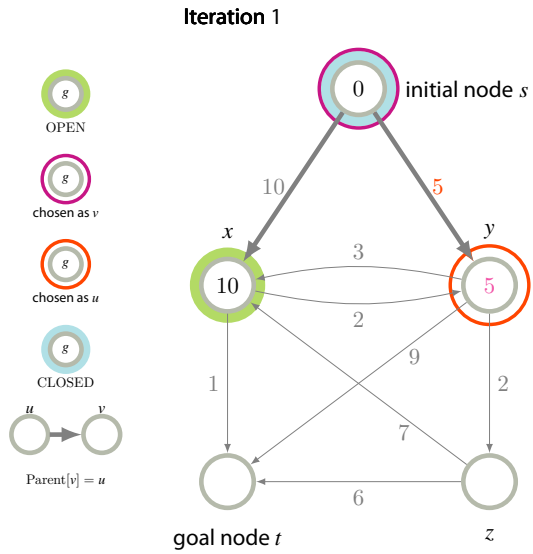
```

1 function Dijkstra(s)
2   OPEN ← new PriorityQueueg
3   g[s] ← 0
4   Insertg(OPEN, s)
5   CLOSED ← ∅
6   loop do
7     if IsEmpty(OPEN) then
8       return "failure"
9     v ← DeleteMing(OPEN)
10    CLOSED ← CLOSED ∪ {v}
11    if IsGoal(v) then
12      return Solution(v, s)
13    Expand(v)
14  procedure Expand(v)
15  foreach u ∈ Succ(v) do
16    if u ∉ OPEN ∪ CLOSED then
17      g[u] ← g[v] + c(v, u)
18      Parent[u] ← v
19      Insertg(OPEN, u)
20    else if u ∈ OPEN then
21      if g[v] + c(v, u) < g[u] then
22        g[u] ← g[v] + c(v, u)
23        Parent[u] ← v
  
```



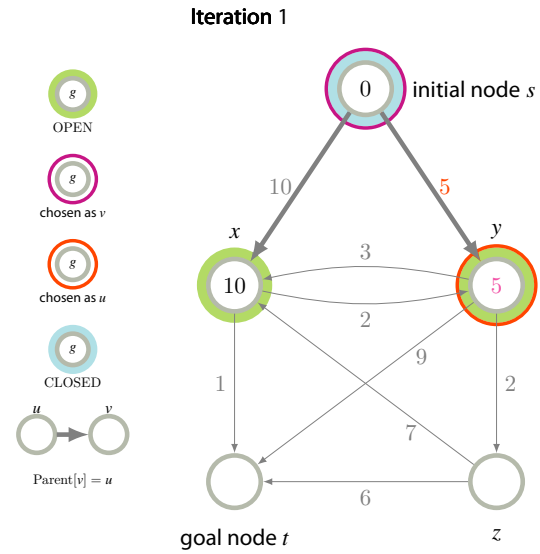
```

1 function Dijkstra(s)
2   OPEN ← new PriorityQueueg
3   g[s] ← 0
4   Insertg(OPEN, s)
5   CLOSED ← ∅
6   loop do
7     if IsEmpty(OPEN) then
8       return "failure"
9     v ← DeleteMing(OPEN)
10    CLOSED ← CLOSED ∪ {v}
11    if IsGoal(v) then
12      return Solution(v, s)
13    Expand(v)
14  procedure Expand(v)
15  foreach u ∈ Succ(v) do
16    if u ∉ OPEN ∪ CLOSED then
17      g[u] ← g[v] + c(v, u)
18      Parent[u] ← v
19      Insertg(OPEN, u)
20    else if u ∈ OPEN then
21      if g[v] + c(v, u) < g[u] then
22        g[u] ← g[v] + c(v, u)
23        Parent[u] ← v
  
```



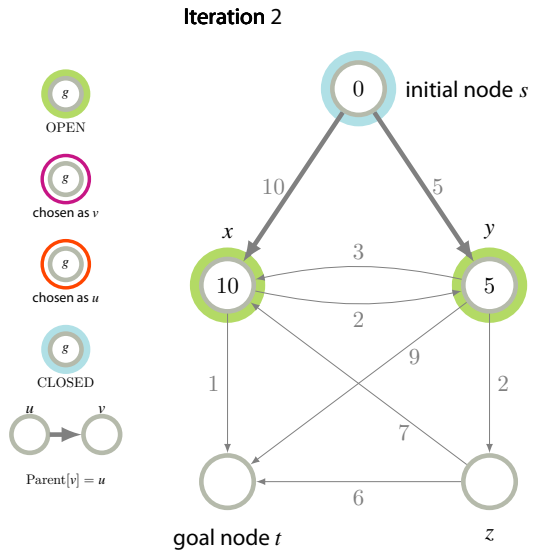
```

1 function Dijkstra(s)
2   OPEN ← new PriorityQueueg
3   g[s] ← 0
4   Insertg(OPEN, s)
5   CLOSED ← ∅
6   loop do
7     if IsEmpty(OPEN) then
8       return "failure"
9     v ← DeleteMing(OPEN)
10    CLOSED ← CLOSED ∪ {v}
11    if IsGoal(v) then
12      return Solution(v, s)
13    Expand(v)
14  procedure Expand(v)
15  foreach u ∈ Succ(v) do
16    if u ∉ OPEN ∪ CLOSED then
17      g[u] ← g[v] + c(v, u)
18      Parent[u] ← v
19      Insertg(OPEN, u)
20    else if u ∈ OPEN then
21      if g[v] + c(v, u) < g[u] then
22        g[u] ← g[v] + c(v, u)
23        Parent[u] ← v
  
```



```

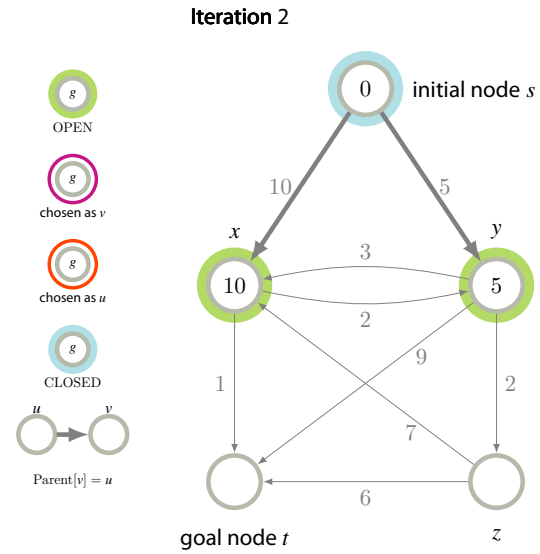
1 function Dijkstra(s)
2   OPEN ← new PriorityQueueg
3   g[s] ← 0
4   Insertg(OPEN, s)
5   CLOSED ← ∅
6   loop do
7     if IsEmpty(OPEN) then
8       return "failure"
9     v ← DeleteMing(OPEN)
10    CLOSED ← CLOSED ∪ {v}
11    if IsGoal(v) then
12      return Solution(v, s)
13    Expand(v)
14  procedure Expand(v)
15  foreach u ∈ Succ(v) do
16    if u ∉ OPEN ∪ CLOSED then
17      g[u] ← g[v] + c(v, u)
18      Parent[u] ← v
19      Insertg(OPEN, u)
20    else if u ∈ OPEN then
21      if g[v] + c(v, u) < g[u] then
22        g[u] ← g[v] + c(v, u)
23        Parent[u] ← v
  
```



```

1 function Dijkstra(s)
2 OPEN ← new PriorityQueueg
3 g[s] ← 0
4 Insertg(OPEN, s)
5 CLOSED ← ∅
6 loop do
7   if IsEmpty(OPEN) then
8     return "failure"
9   v ← DeleteMing(OPEN)
10  CLOSED ← CLOSED ∪ {v}
11  if IsGoal(v) then
12    return Solution(v, s)
13  Expand(v)
14 procedure Expand(v)
15 foreach u ∈ Succ(v) do
16   if u ∉ OPEN ∪ CLOSED then
17     g[u] ← g[v] + c(v, u)
18     Parent[u] ← v
19     Insertg(OPEN, u)
20   else if u ∈ OPEN then
21     if g[v] + c(v, u) < g[u] then
22       g[u] ← g[v] + c(v, u)
23       Parent[u] ← v

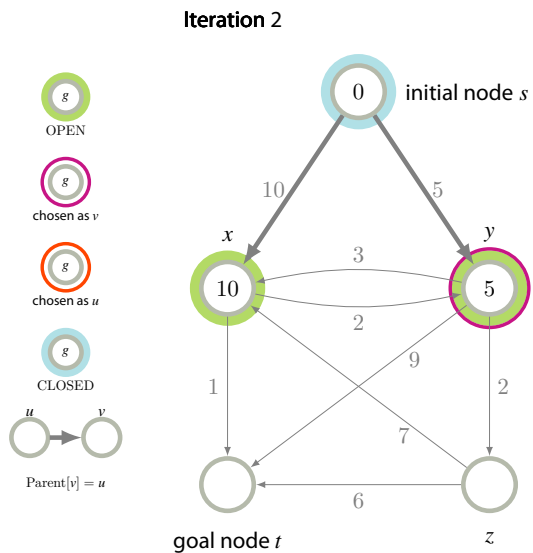
```



```

1 function Dijkstra(s)
2 OPEN ← new PriorityQueueg
3 g[s] ← 0
4 Insertg(OPEN, s)
5 CLOSED ← ∅
6 loop do
7   if IsEmpty(OPEN) then
8     return "failure"
9   v ← DeleteMing(OPEN)
10  CLOSED ← CLOSED ∪ {v}
11  if IsGoal(v) then
12    return Solution(v, s)
13  Expand(v)
14 procedure Expand(v)
15 foreach u ∈ Succ(v) do
16   if u ∉ OPEN ∪ CLOSED then
17     g[u] ← g[v] + c(v, u)
18     Parent[u] ← v
19     Insertg(OPEN, u)
20   else if u ∈ OPEN then
21     if g[v] + c(v, u) < g[u] then
22       g[u] ← g[v] + c(v, u)
23       Parent[u] ← v

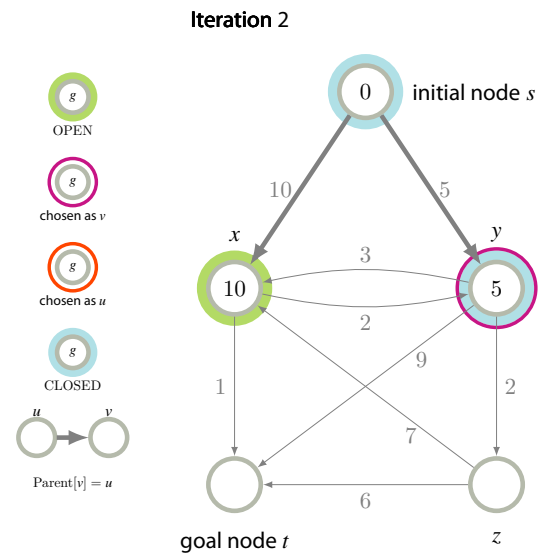
```



```

1 function Dijkstra(s)
2 OPEN ← new PriorityQueueg
3 g[s] ← 0
4 Insertg(OPEN, s)
5 CLOSED ← ∅
6 loop do
7   if IsEmpty(OPEN) then
8     return "failure"
9   v ← DeleteMing(OPEN)
10  CLOSED ← CLOSED ∪ {v}
11  if IsGoal(v) then
12    return Solution(v, s)
13  Expand(v)
14 procedure Expand(v)
15 foreach u ∈ Succ(v) do
16   if u ∉ OPEN ∪ CLOSED then
17     g[u] ← g[v] + c(v, u)
18     Parent[u] ← v
19     Insertg(OPEN, u)
20   else if u ∈ OPEN then
21     if g[v] + c(v, u) < g[u] then
22       g[u] ← g[v] + c(v, u)
23       Parent[u] ← v

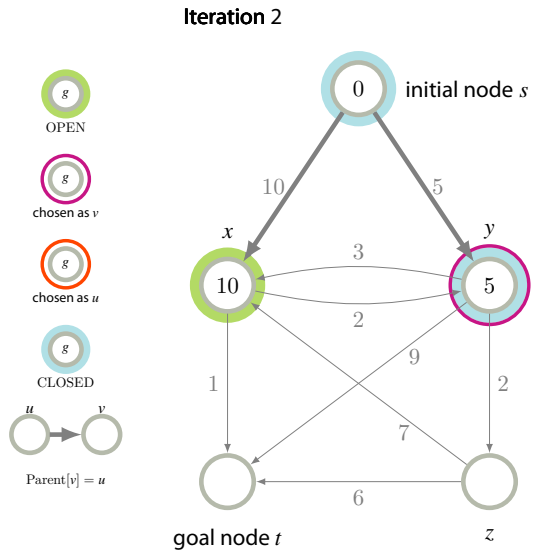
```



```

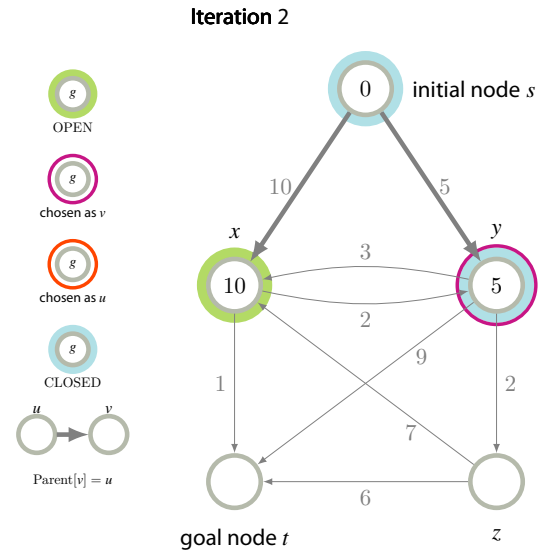
1 function Dijkstra(s)
2 OPEN ← new PriorityQueueg
3 g[s] ← 0
4 Insertg(OPEN, s)
5 CLOSED ← ∅
6 loop do
7   if IsEmpty(OPEN) then
8     return "failure"
9   v ← DeleteMing(OPEN)
10  CLOSED ← CLOSED ∪ {v}
11  if IsGoal(v) then
12    return Solution(v, s)
13  Expand(v)
14 procedure Expand(v)
15 foreach u ∈ Succ(v) do
16   if u ∉ OPEN ∪ CLOSED then
17     g[u] ← g[v] + c(v, u)
18     Parent[u] ← v
19     Insertg(OPEN, u)
20   else if u ∈ OPEN then
21     if g[v] + c(v, u) < g[u] then
22       g[u] ← g[v] + c(v, u)
23       Parent[u] ← v

```



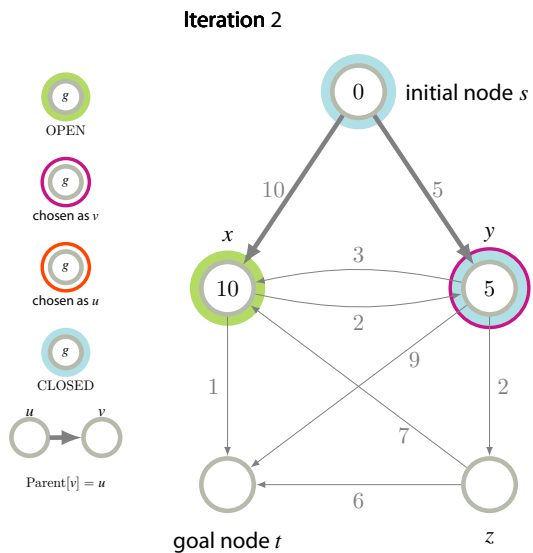
```

1 function Dijkstra(s)
2 OPEN ← new PriorityQueueg
3 g[s] ← 0
4 Insertg(OPEN, s)
5 CLOSED ← ∅
6 loop do
7   if IsEmpty(OPEN) then
8     return "failure"
9   v ← DeleteMing(OPEN)
10  CLOSED ← CLOSED ∪ {v}
11  if IsGoal(v) then
12    return Solution(v, s)
13  Expand(v)
14 procedure Expand(v)
15 foreach u ∈ Succ(v) do
16   if u ∉ OPEN ∪ CLOSED then
17     g[u] ← g[v] + c(v, u)
18     Parent[u] ← v
19     Insertg(OPEN, u)
20   else if u ∈ OPEN then
21     if g[v] + c(v, u) < g[u] then
22       g[u] ← g[v] + c(v, u)
23       Parent[u] ← v
  
```



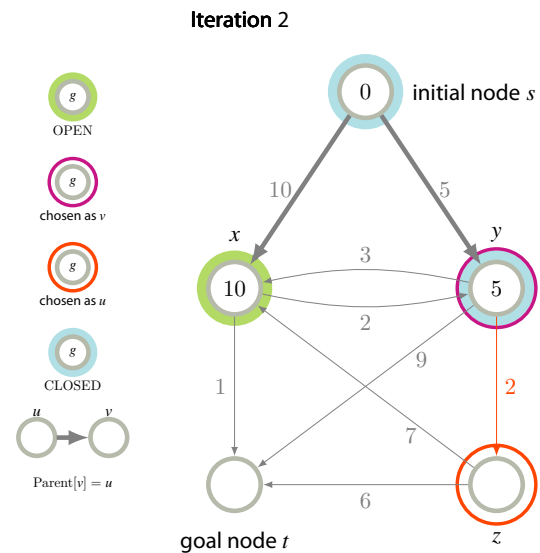
```

1 function Dijkstra(s)
2 OPEN ← new PriorityQueueg
3 g[s] ← 0
4 Insertg(OPEN, s)
5 CLOSED ← ∅
6 loop do
7   if IsEmpty(OPEN) then
8     return "failure"
9   v ← DeleteMing(OPEN)
10  CLOSED ← CLOSED ∪ {v}
11  if IsGoal(v) then
12    return Solution(v, s)
13  Expand(v)
14 procedure Expand(v)
15 foreach u ∈ Succ(v) do
16   if u ∉ OPEN ∪ CLOSED then
17     g[u] ← g[v] + c(v, u)
18     Parent[u] ← v
19     Insertg(OPEN, u)
20   else if u ∈ OPEN then
21     if g[v] + c(v, u) < g[u] then
22       g[u] ← g[v] + c(v, u)
23       Parent[u] ← v
  
```



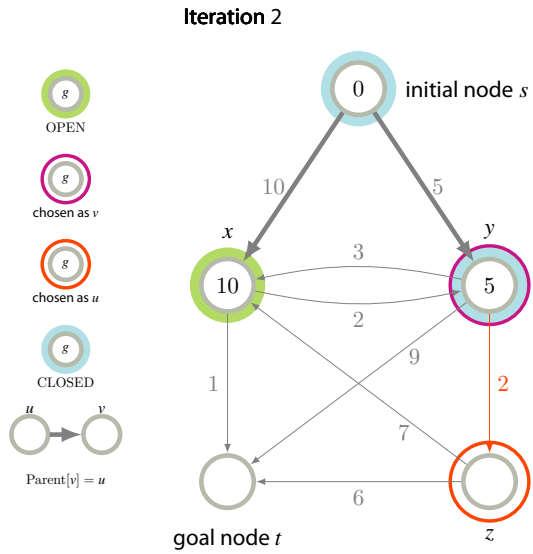
```

1 function Dijkstra(s)
2 OPEN ← new PriorityQueueg
3 g[s] ← 0
4 Insertg(OPEN, s)
5 CLOSED ← ∅
6 loop do
7   if IsEmpty(OPEN) then
8     return "failure"
9   v ← DeleteMing(OPEN)
10  CLOSED ← CLOSED ∪ {v}
11  if IsGoal(v) then
12    return Solution(v, s)
13  Expand(v)
14 procedure Expand(v)
15 foreach u ∈ Succ(v) do
16   if u ∉ OPEN ∪ CLOSED then
17     g[u] ← g[v] + c(v, u)
18     Parent[u] ← v
19     Insertg(OPEN, u)
20   else if u ∈ OPEN then
21     if g[v] + c(v, u) < g[u] then
22       g[u] ← g[v] + c(v, u)
23       Parent[u] ← v
  
```



```

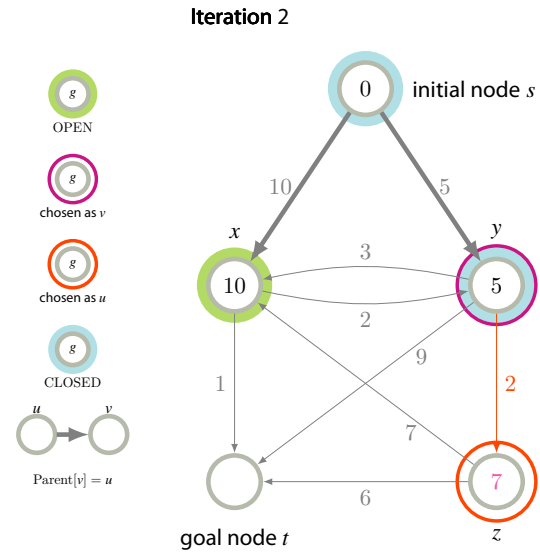
1 function Dijkstra(s)
2 OPEN ← new PriorityQueueg
3 g[s] ← 0
4 Insertg(OPEN, s)
5 CLOSED ← ∅
6 loop do
7   if IsEmpty(OPEN) then
8     return "failure"
9   v ← DeleteMing(OPEN)
10  CLOSED ← CLOSED ∪ {v}
11  if IsGoal(v) then
12    return Solution(v, s)
13  Expand(v)
14 procedure Expand(v)
15 foreach u ∈ Succ(v) do
16   if u ∉ OPEN ∪ CLOSED then
17     g[u] ← g[v] + c(v, u)
18     Parent[u] ← v
19     Insertg(OPEN, u)
20   else if u ∈ OPEN then
21     if g[v] + c(v, u) < g[u] then
22       g[u] ← g[v] + c(v, u)
23       Parent[u] ← v
  
```



```

1 function Dijkstra( $s$ )
2 OPEN  $\leftarrow$  new PriorityQueue $_g$ 
3  $g[s] \leftarrow 0$ 
4 Insert $_g$ (OPEN,  $s$ )
5 CLOSED  $\leftarrow \emptyset$ 
6 loop do
7   if IsEmpty(OPEN) then
8     return "failure"
9    $v \leftarrow$  DeleteMin $_g$ (OPEN)
10  CLOSED  $\leftarrow$  CLOSED  $\cup \{v\}$ 
11  if IsGoal( $v$ ) then
12    return Solution( $v, s$ )
13  Expand( $v$ )
14 procedure Expand( $v$ )
15 foreach  $u \in$  Succ( $v$ ) do
16   if  $u \notin$  OPEN  $\cup$  CLOSED then
17      $g[u] \leftarrow g[v] + c(v, u)$ 
18     Parent[ $u$ ]  $\leftarrow v$ 
19     Insert $_g$ (OPEN,  $u$ )
20   else if  $u \in$  OPEN then
21     if  $g[v] + c(v, u) < g[u]$  then
22        $g[u] \leftarrow g[v] + c(v, u)$ 
23       Parent[ $u$ ]  $\leftarrow v$ 

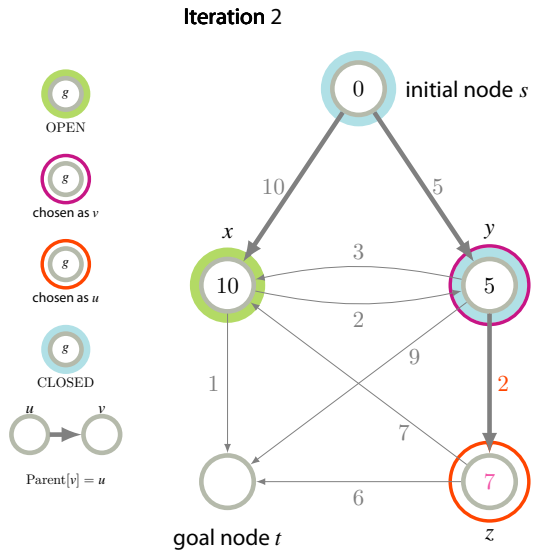
```



```

1 function Dijkstra( $s$ )
2 OPEN  $\leftarrow$  new PriorityQueue $_g$ 
3  $g[s] \leftarrow 0$ 
4 Insert $_g$ (OPEN,  $s$ )
5 CLOSED  $\leftarrow \emptyset$ 
6 loop do
7   if IsEmpty(OPEN) then
8     return "failure"
9    $v \leftarrow$  DeleteMin $_g$ (OPEN)
10  CLOSED  $\leftarrow$  CLOSED  $\cup \{v\}$ 
11  if IsGoal( $v$ ) then
12    return Solution( $v, s$ )
13  Expand( $v$ )
14 procedure Expand( $v$ )
15 foreach  $u \in$  Succ( $v$ ) do
16   if  $u \notin$  OPEN  $\cup$  CLOSED then
17      $g[u] \leftarrow g[v] + c(v, u)$ 
18     Parent[ $u$ ]  $\leftarrow v$ 
19     Insert $_g$ (OPEN,  $u$ )
20   else if  $u \in$  OPEN then
21     if  $g[v] + c(v, u) < g[u]$  then
22        $g[u] \leftarrow g[v] + c(v, u)$ 
23       Parent[ $u$ ]  $\leftarrow v$ 

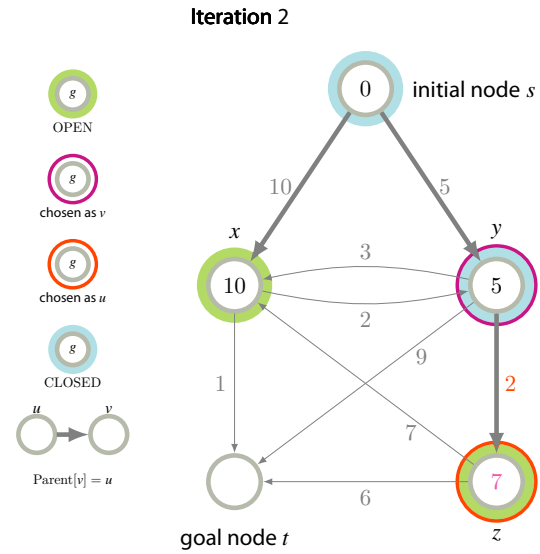
```



```

1 function Dijkstra( $s$ )
2 OPEN  $\leftarrow$  new PriorityQueue $_g$ 
3  $g[s] \leftarrow 0$ 
4 Insert $_g$ (OPEN,  $s$ )
5 CLOSED  $\leftarrow \emptyset$ 
6 loop do
7   if IsEmpty(OPEN) then
8     return "failure"
9    $v \leftarrow$  DeleteMin $_g$ (OPEN)
10  CLOSED  $\leftarrow$  CLOSED  $\cup \{v\}$ 
11  if IsGoal( $v$ ) then
12    return Solution( $v, s$ )
13  Expand( $v$ )
14 procedure Expand( $v$ )
15 foreach  $u \in$  Succ( $v$ ) do
16   if  $u \notin$  OPEN  $\cup$  CLOSED then
17      $g[u] \leftarrow g[v] + c(v, u)$ 
18     Parent[ $u$ ]  $\leftarrow v$ 
19     Insert $_g$ (OPEN,  $u$ )
20   else if  $u \in$  OPEN then
21     if  $g[v] + c(v, u) < g[u]$  then
22        $g[u] \leftarrow g[v] + c(v, u)$ 
23       Parent[ $u$ ]  $\leftarrow v$ 

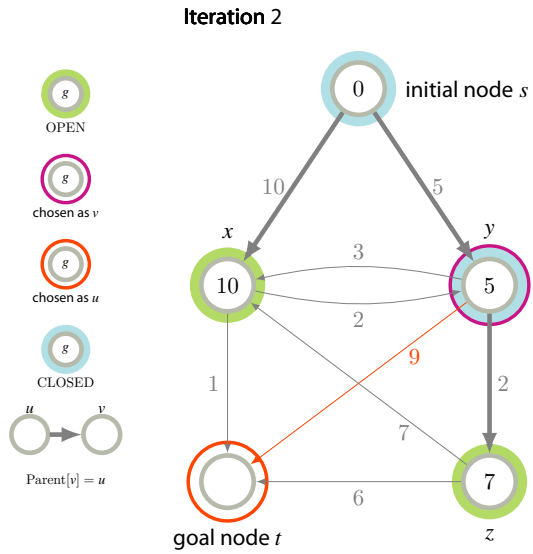
```



```

1 function Dijkstra( $s$ )
2 OPEN  $\leftarrow$  new PriorityQueue $_g$ 
3  $g[s] \leftarrow 0$ 
4 Insert $_g$ (OPEN,  $s$ )
5 CLOSED  $\leftarrow \emptyset$ 
6 loop do
7   if IsEmpty(OPEN) then
8     return "failure"
9    $v \leftarrow$  DeleteMin $_g$ (OPEN)
10  CLOSED  $\leftarrow$  CLOSED  $\cup \{v\}$ 
11  if IsGoal( $v$ ) then
12    return Solution( $v, s$ )
13  Expand( $v$ )
14 procedure Expand( $v$ )
15 foreach  $u \in$  Succ( $v$ ) do
16   if  $u \notin$  OPEN  $\cup$  CLOSED then
17      $g[u] \leftarrow g[v] + c(v, u)$ 
18     Parent[ $u$ ]  $\leftarrow v$ 
19     Insert $_g$ (OPEN,  $u$ )
20   else if  $u \in$  OPEN then
21     if  $g[v] + c(v, u) < g[u]$  then
22        $g[u] \leftarrow g[v] + c(v, u)$ 
23       Parent[ $u$ ]  $\leftarrow v$ 

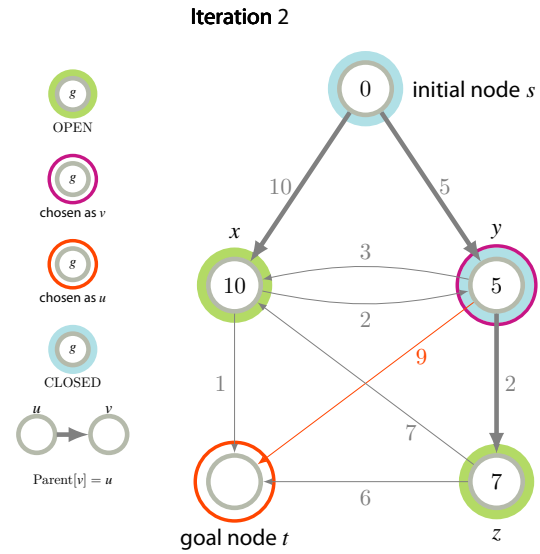
```

```

1 function Dijkstra(s)
2 OPEN ← new PriorityQueueg
3 g[s] ← 0
4 Insertg(OPEN, s)
5 CLOSED ← ∅
6 loop do
7   if IsEmpty(OPEN) then
8     return "failure"
9   v ← DeleteMing(OPEN)
10  CLOSED ← CLOSED ∪ {v}
11  if IsGoal(v) then
12    return Solution(v, s)
13  Expand(v)
14 procedure Expand(v)
15 foreach u ∈ Succ(v) do
16   if u ∉ OPEN ∪ CLOSED then
17     g[u] ← g[v] + c(v, u)
18     Parent[u] ← v
19     Insertg(OPEN, u)
20   else if u ∈ OPEN then
21     if g[v] + c(v, u) < g[u] then
22       g[u] ← g[v] + c(v, u)
23       Parent[u] ← v

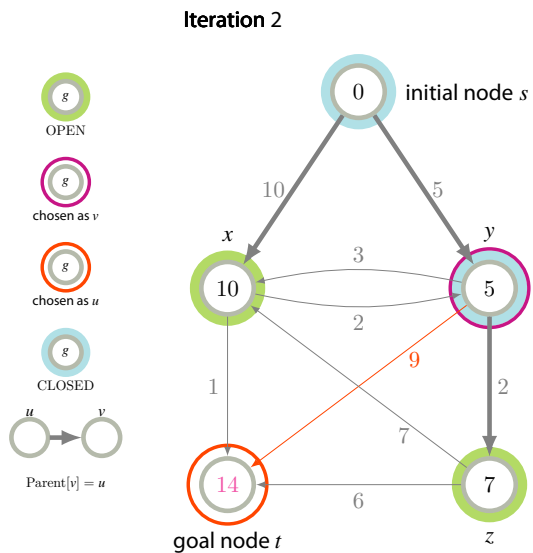
```



```

1 function Dijkstra(s)
2 OPEN ← new PriorityQueueg
3 g[s] ← 0
4 Insertg(OPEN, s)
5 CLOSED ← ∅
6 loop do
7   if IsEmpty(OPEN) then
8     return "failure"
9   v ← DeleteMing(OPEN)
10  CLOSED ← CLOSED ∪ {v}
11  if IsGoal(v) then
12    return Solution(v, s)
13  Expand(v)
14 procedure Expand(v)
15 foreach u ∈ Succ(v) do
16   if u ∉ OPEN ∪ CLOSED then
17     g[u] ← g[v] + c(v, u)
18     Parent[u] ← v
19     Insertg(OPEN, u)
20   else if u ∈ OPEN then
21     if g[v] + c(v, u) < g[u] then
22       g[u] ← g[v] + c(v, u)
23       Parent[u] ← v

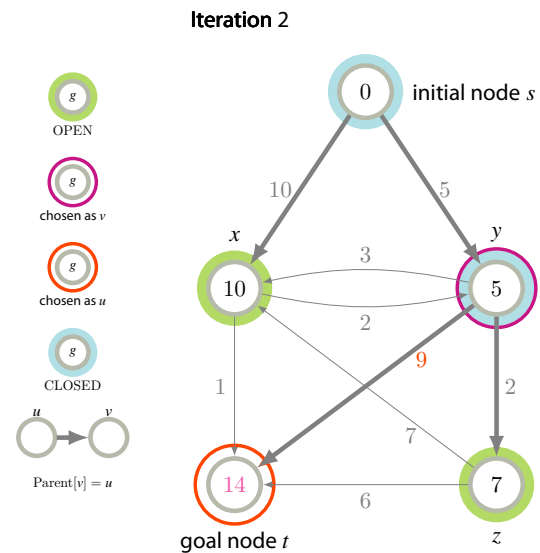
```



```

1 function Dijkstra(s)
2 OPEN ← new PriorityQueueg
3 g[s] ← 0
4 Insertg(OPEN, s)
5 CLOSED ← ∅
6 loop do
7   if IsEmpty(OPEN) then
8     return "failure"
9   v ← DeleteMing(OPEN)
10  CLOSED ← CLOSED ∪ {v}
11  if IsGoal(v) then
12    return Solution(v, s)
13  Expand(v)
14 procedure Expand(v)
15 foreach u ∈ Succ(v) do
16   if u ∉ OPEN ∪ CLOSED then
17     g[u] ← g[v] + c(v, u)
18     Parent[u] ← v
19     Insertg(OPEN, u)
20   else if u ∈ OPEN then
21     if g[v] + c(v, u) < g[u] then
22       g[u] ← g[v] + c(v, u)
23       Parent[u] ← v

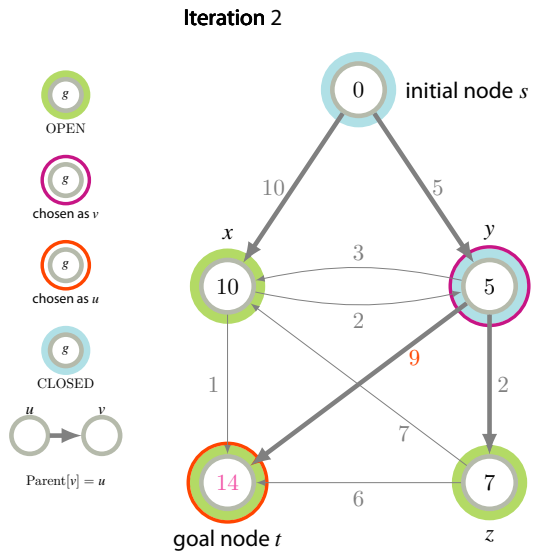
```



```

1 function Dijkstra(s)
2 OPEN ← new PriorityQueueg
3 g[s] ← 0
4 Insertg(OPEN, s)
5 CLOSED ← ∅
6 loop do
7   if IsEmpty(OPEN) then
8     return "failure"
9   v ← DeleteMing(OPEN)
10  CLOSED ← CLOSED ∪ {v}
11  if IsGoal(v) then
12    return Solution(v, s)
13  Expand(v)
14 procedure Expand(v)
15 foreach u ∈ Succ(v) do
16   if u ∉ OPEN ∪ CLOSED then
17     g[u] ← g[v] + c(v, u)
18     Parent[u] ← v
19     Insertg(OPEN, u)
20   else if u ∈ OPEN then
21     if g[v] + c(v, u) < g[u] then
22       g[u] ← g[v] + c(v, u)
23       Parent[u] ← v

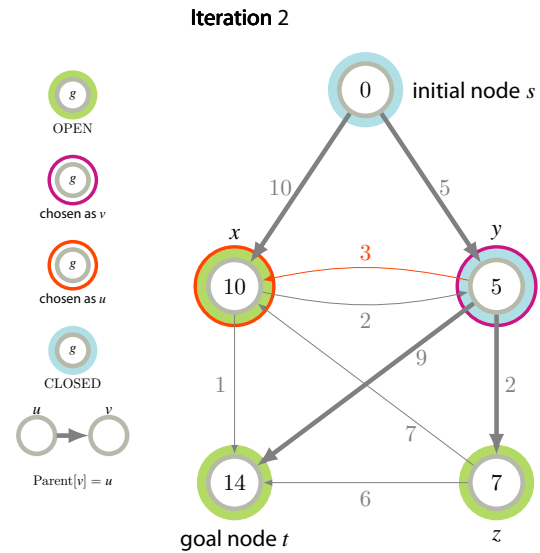
```



```

1 function Dijkstra(s)
2 OPEN ← new PriorityQueueg
3 g[s] ← 0
4 Insertg(OPEN, s)
5 CLOSED ← ∅
6 loop do
7   if IsEmpty(OPEN) then
8     return "failure"
9   v ← DeleteMing(OPEN)
10  CLOSED ← CLOSED ∪ {v}
11  if IsGoal(v) then
12    return Solution(v, s)
13  Expand(v)
14 procedure Expand(v)
15 foreach u ∈ Succ(v) do
16   if u ∉ OPEN ∪ CLOSED then
17     g[u] ← g[v] + c(v, u)
18     Parent[u] ← v
19     Insertg(OPEN, u)
20   else if u ∈ OPEN then
21     if g[v] + c(v, u) < g[u] then
22       g[u] ← g[v] + c(v, u)
23       Parent[u] ← v

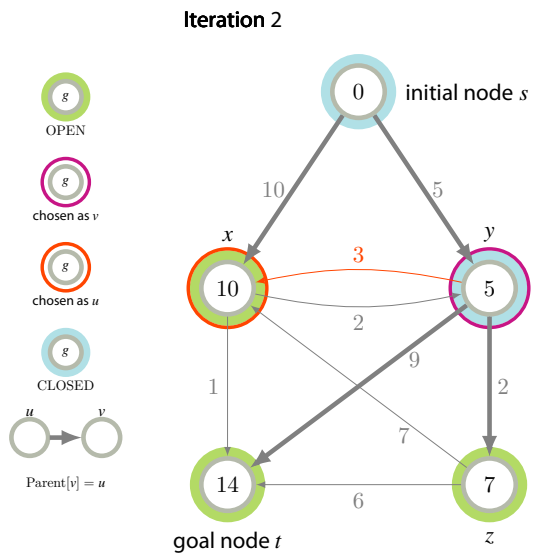
```



```

1 function Dijkstra(s)
2 OPEN ← new PriorityQueueg
3 g[s] ← 0
4 Insertg(OPEN, s)
5 CLOSED ← ∅
6 loop do
7   if IsEmpty(OPEN) then
8     return "failure"
9   v ← DeleteMing(OPEN)
10  CLOSED ← CLOSED ∪ {v}
11  if IsGoal(v) then
12    return Solution(v, s)
13  Expand(v)
14 procedure Expand(v)
15 foreach u ∈ Succ(v) do
16   if u ∉ OPEN ∪ CLOSED then
17     g[u] ← g[v] + c(v, u)
18     Parent[u] ← v
19     Insertg(OPEN, u)
20   else if u ∈ OPEN then
21     if g[v] + c(v, u) < g[u] then
22       g[u] ← g[v] + c(v, u)
23       Parent[u] ← v

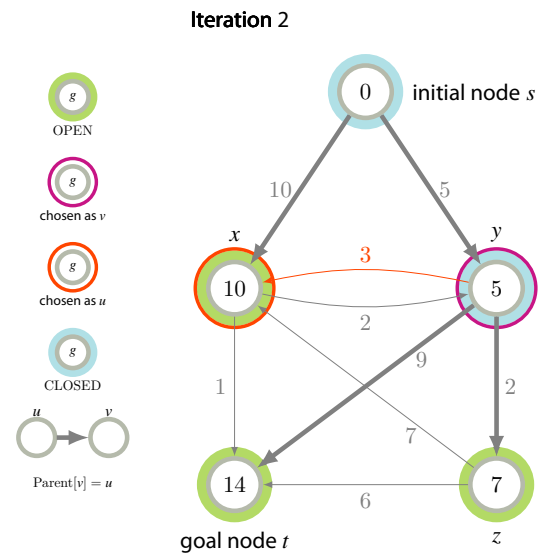
```



```

1 function Dijkstra(s)
2 OPEN ← new PriorityQueueg
3 g[s] ← 0
4 Insertg(OPEN, s)
5 CLOSED ← ∅
6 loop do
7   if IsEmpty(OPEN) then
8     return "failure"
9   v ← DeleteMing(OPEN)
10  CLOSED ← CLOSED ∪ {v}
11  if IsGoal(v) then
12    return Solution(v, s)
13  Expand(v)
14 procedure Expand(v)
15 foreach u ∈ Succ(v) do
16   if u ∉ OPEN ∪ CLOSED then
17     g[u] ← g[v] + c(v, u)
18     Parent[u] ← v
19     Insertg(OPEN, u)
20   else if u ∈ OPEN then
21     if g[v] + c(v, u) < g[u] then
22       g[u] ← g[v] + c(v, u)
23       Parent[u] ← v

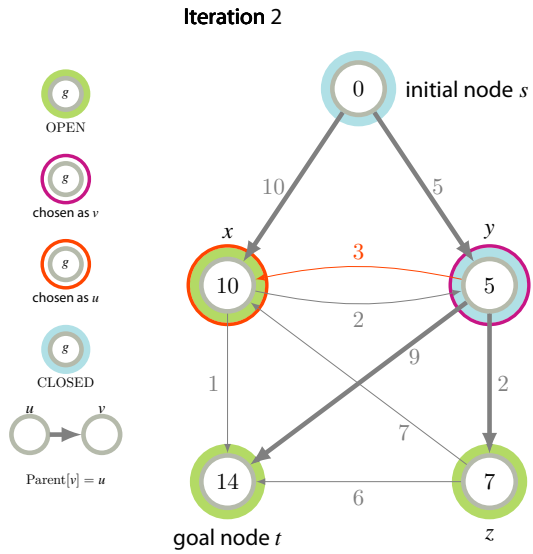
```



```

1 function Dijkstra(s)
2 OPEN ← new PriorityQueueg
3 g[s] ← 0
4 Insertg(OPEN, s)
5 CLOSED ← ∅
6 loop do
7   if IsEmpty(OPEN) then
8     return "failure"
9   v ← DeleteMing(OPEN)
10  CLOSED ← CLOSED ∪ {v}
11  if IsGoal(v) then
12    return Solution(v, s)
13  Expand(v)
14 procedure Expand(v)
15 foreach u ∈ Succ(v) do
16   if u ∉ OPEN ∪ CLOSED then
17     g[u] ← g[v] + c(v, u)
18     Parent[u] ← v
19     Insertg(OPEN, u)
20   else if u ∈ OPEN then
21     if g[v] + c(v, u) < g[u] then
22       g[u] ← g[v] + c(v, u)
23       Parent[u] ← v

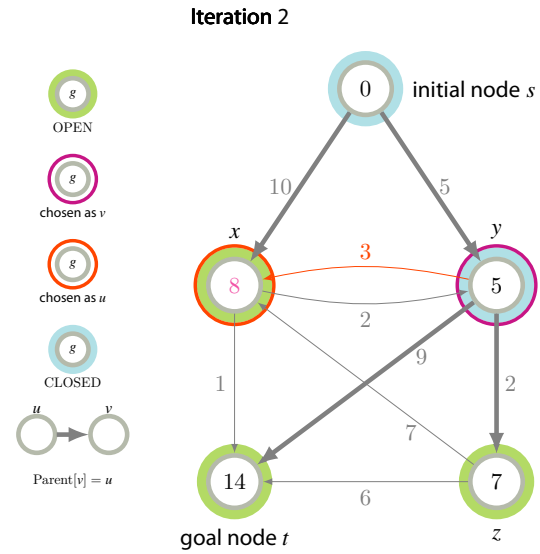
```



```

1 function Dijkstra(s)
2   OPEN ← new PriorityQueueg
3   g[s] ← 0
4   Insertg(OPEN, s)
5   CLOSED ← ∅
6   loop do
7     if IsEmpty(OPEN) then
8       return "failure"
9     v ← DeleteMing(OPEN)
10    CLOSED ← CLOSED ∪ {v}
11    if IsGoal(v) then
12      return Solution(v, s)
13    Expand(v)
14  procedure Expand(v)
15  foreach u ∈ Succ(v) do
16    if u ∉ OPEN ∪ CLOSED then
17      g[u] ← g[v] + c(v, u)
18      Parent[u] ← v
19      Insertg(OPEN, u)
20    else if u ∈ OPEN then
21      if g[v] + c(v, u) < g[u] then
22        g[u] ← g[v] + c(v, u)
23        Parent[u] ← v

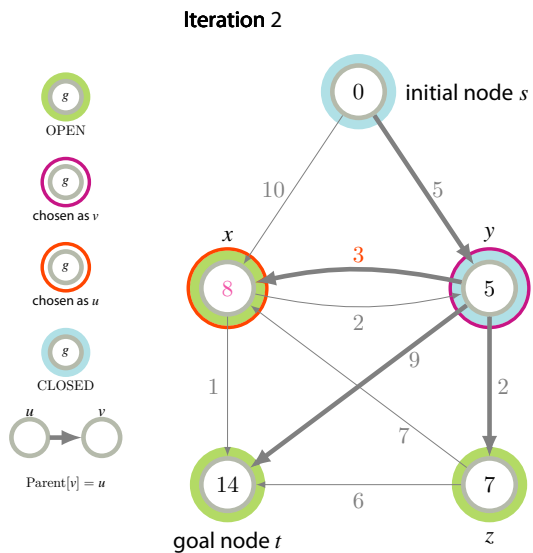
```



```

1 function Dijkstra(s)
2   OPEN ← new PriorityQueueg
3   g[s] ← 0
4   Insertg(OPEN, s)
5   CLOSED ← ∅
6   loop do
7     if IsEmpty(OPEN) then
8       return "failure"
9     v ← DeleteMing(OPEN)
10    CLOSED ← CLOSED ∪ {v}
11    if IsGoal(v) then
12      return Solution(v, s)
13    Expand(v)
14  procedure Expand(v)
15  foreach u ∈ Succ(v) do
16    if u ∉ OPEN ∪ CLOSED then
17      g[u] ← g[v] + c(v, u)
18      Parent[u] ← v
19      Insertg(OPEN, u)
20    else if u ∈ OPEN then
21      if g[v] + c(v, u) < g[u] then
22        g[u] ← g[v] + c(v, u)
23        Parent[u] ← v

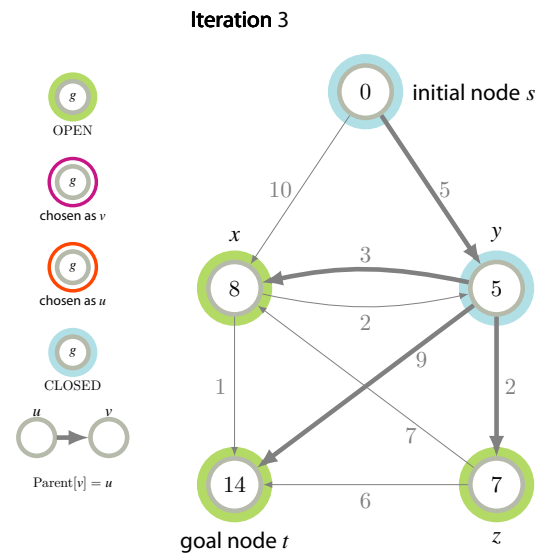
```



```

1 function Dijkstra(s)
2   OPEN ← new PriorityQueueg
3   g[s] ← 0
4   Insertg(OPEN, s)
5   CLOSED ← ∅
6   loop do
7     if IsEmpty(OPEN) then
8       return "failure"
9     v ← DeleteMing(OPEN)
10    CLOSED ← CLOSED ∪ {v}
11    if IsGoal(v) then
12      return Solution(v, s)
13    Expand(v)
14  procedure Expand(v)
15  foreach u ∈ Succ(v) do
16    if u ∉ OPEN ∪ CLOSED then
17      g[u] ← g[v] + c(v, u)
18      Parent[u] ← v
19      Insertg(OPEN, u)
20    else if u ∈ OPEN then
21      if g[v] + c(v, u) < g[u] then
22        g[u] ← g[v] + c(v, u)
23        Parent[u] ← v

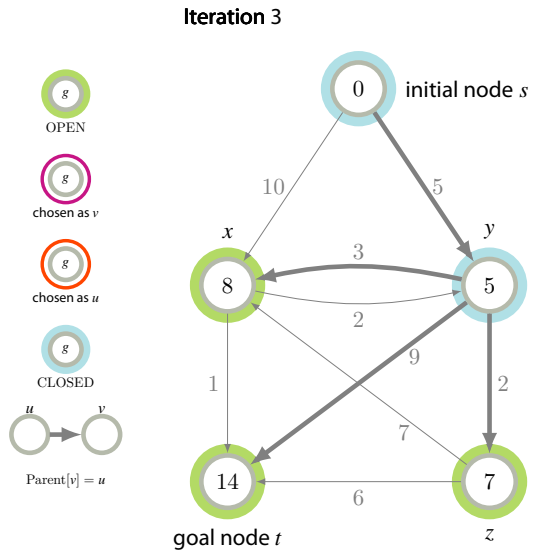
```



```

1 function Dijkstra(s)
2   OPEN ← new PriorityQueueg
3   g[s] ← 0
4   Insertg(OPEN, s)
5   CLOSED ← ∅
6   loop do
7     if IsEmpty(OPEN) then
8       return "failure"
9     v ← DeleteMing(OPEN)
10    CLOSED ← CLOSED ∪ {v}
11    if IsGoal(v) then
12      return Solution(v, s)
13    Expand(v)
14  procedure Expand(v)
15  foreach u ∈ Succ(v) do
16    if u ∉ OPEN ∪ CLOSED then
17      g[u] ← g[v] + c(v, u)
18      Parent[u] ← v
19      Insertg(OPEN, u)
20    else if u ∈ OPEN then
21      if g[v] + c(v, u) < g[u] then
22        g[u] ← g[v] + c(v, u)
23        Parent[u] ← v

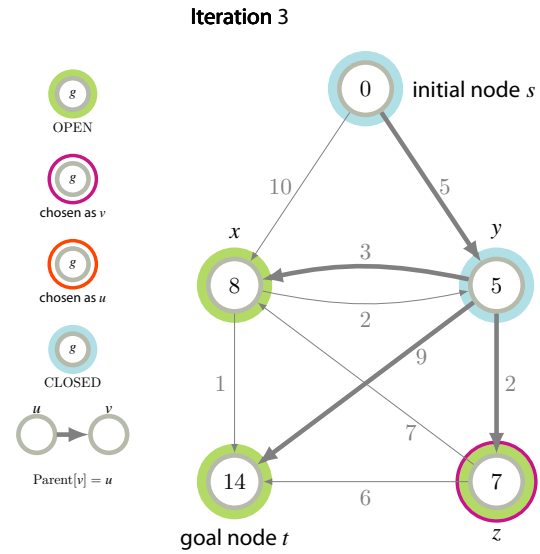
```



```

1 function Dijkstra(s)
2   OPEN ← new PriorityQueueg
3   g[s] ← 0
4   Insertg(OPEN, s)
5   CLOSED ← ∅
6   loop do
7     if IsEmpty(OPEN) then
8       return "failure"
9     v ← DeleteMing(OPEN)
10    CLOSED ← CLOSED ∪ {v}
11    if IsGoal(v) then
12      return Solution(v, s)
13    Expand(v)
14  procedure Expand(v)
15  foreach u ∈ Succ(v) do
16    if u ∉ OPEN ∪ CLOSED then
17      g[u] ← g[v] + c(v, u)
18      Parent[u] ← v
19      Insertg(OPEN, u)
20    else if u ∈ OPEN then
21      if g[v] + c(v, u) < g[u] then
22        g[u] ← g[v] + c(v, u)
23        Parent[u] ← v

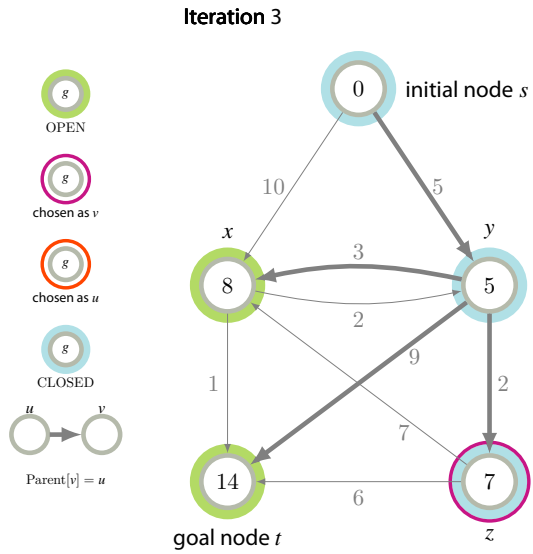
```



```

1 function Dijkstra(s)
2   OPEN ← new PriorityQueueg
3   g[s] ← 0
4   Insertg(OPEN, s)
5   CLOSED ← ∅
6   loop do
7     if IsEmpty(OPEN) then
8       return "failure"
9     v ← DeleteMing(OPEN)
10    CLOSED ← CLOSED ∪ {v}
11    if IsGoal(v) then
12      return Solution(v, s)
13    Expand(v)
14  procedure Expand(v)
15  foreach u ∈ Succ(v) do
16    if u ∉ OPEN ∪ CLOSED then
17      g[u] ← g[v] + c(v, u)
18      Parent[u] ← v
19      Insertg(OPEN, u)
20    else if u ∈ OPEN then
21      if g[v] + c(v, u) < g[u] then
22        g[u] ← g[v] + c(v, u)
23        Parent[u] ← v

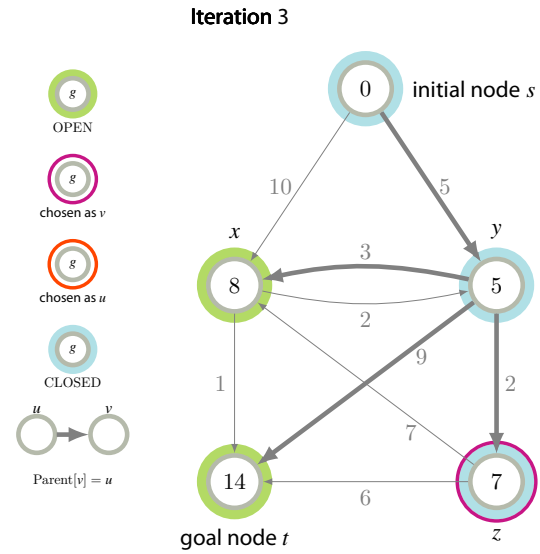
```



```

1 function Dijkstra(s)
2   OPEN ← new PriorityQueueg
3   g[s] ← 0
4   Insertg(OPEN, s)
5   CLOSED ← ∅
6   loop do
7     if IsEmpty(OPEN) then
8       return "failure"
9     v ← DeleteMing(OPEN)
10    CLOSED ← CLOSED ∪ {v}
11    if IsGoal(v) then
12      return Solution(v, s)
13    Expand(v)
14  procedure Expand(v)
15  foreach u ∈ Succ(v) do
16    if u ∉ OPEN ∪ CLOSED then
17      g[u] ← g[v] + c(v, u)
18      Parent[u] ← v
19      Insertg(OPEN, u)
20    else if u ∈ OPEN then
21      if g[v] + c(v, u) < g[u] then
22        g[u] ← g[v] + c(v, u)
23        Parent[u] ← v

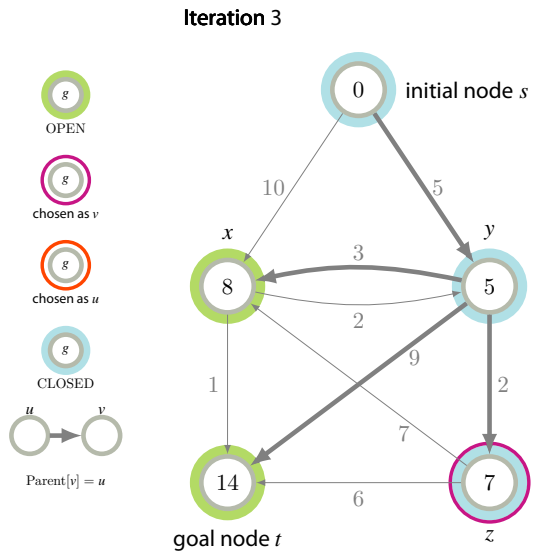
```



```

1 function Dijkstra(s)
2   OPEN ← new PriorityQueueg
3   g[s] ← 0
4   Insertg(OPEN, s)
5   CLOSED ← ∅
6   loop do
7     if IsEmpty(OPEN) then
8       return "failure"
9     v ← DeleteMing(OPEN)
10    CLOSED ← CLOSED ∪ {v}
11    if IsGoal(v) then
12      return Solution(v, s)
13    Expand(v)
14  procedure Expand(v)
15  foreach u ∈ Succ(v) do
16    if u ∉ OPEN ∪ CLOSED then
17      g[u] ← g[v] + c(v, u)
18      Parent[u] ← v
19      Insertg(OPEN, u)
20    else if u ∈ OPEN then
21      if g[v] + c(v, u) < g[u] then
22        g[u] ← g[v] + c(v, u)
23        Parent[u] ← v

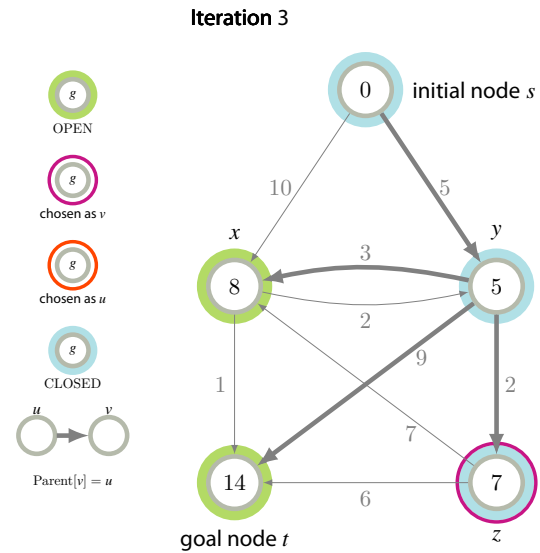
```



```

1 function Dijkstra(s)
2   OPEN ← new PriorityQueueg
3   g[s] ← 0
4   Insertg(OPEN, s)
5   CLOSED ← ∅
6   loop do
7     if IsEmpty(OPEN) then
8       return "failure"
9     v ← DeleteMing(OPEN)
10    CLOSED ← CLOSED ∪ {v}
11    if IsGoal(v) then
12      return Solution(v, s)
13    Expand(v)
14  procedure Expand(v)
15  foreach u ∈ Succ(v) do
16    if u ∉ OPEN ∪ CLOSED then
17      g[u] ← g[v] + c(v, u)
18      Parent[u] ← v
19      Insertg(OPEN, u)
20    else if u ∈ OPEN then
21      if g[v] + c(v, u) < g[u] then
22        g[u] ← g[v] + c(v, u)
23        Parent[u] ← v

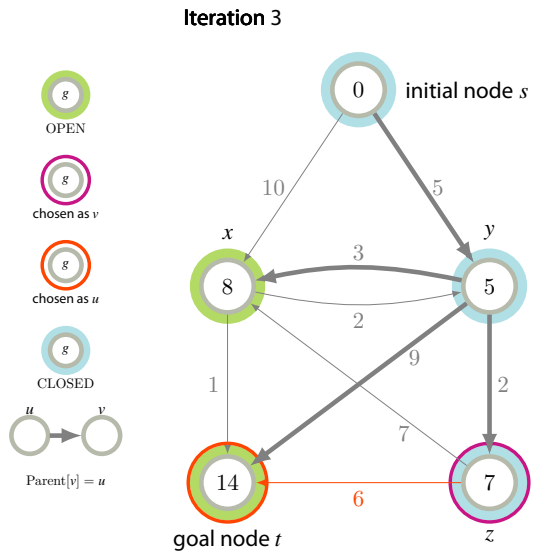
```



```

1 function Dijkstra(s)
2   OPEN ← new PriorityQueueg
3   g[s] ← 0
4   Insertg(OPEN, s)
5   CLOSED ← ∅
6   loop do
7     if IsEmpty(OPEN) then
8       return "failure"
9     v ← DeleteMing(OPEN)
10    CLOSED ← CLOSED ∪ {v}
11    if IsGoal(v) then
12      return Solution(v, s)
13    Expand(v)
14  procedure Expand(v)
15  foreach u ∈ Succ(v) do
16    if u ∉ OPEN ∪ CLOSED then
17      g[u] ← g[v] + c(v, u)
18      Parent[u] ← v
19      Insertg(OPEN, u)
20    else if u ∈ OPEN then
21      if g[v] + c(v, u) < g[u] then
22        g[u] ← g[v] + c(v, u)
23        Parent[u] ← v

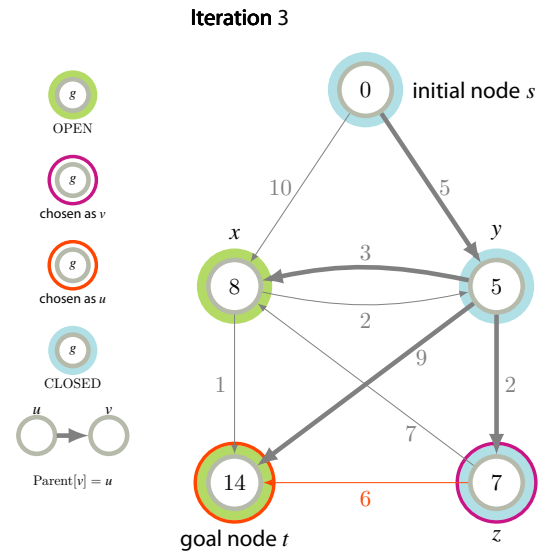
```



```

1 function Dijkstra(s)
2   OPEN ← new PriorityQueueg
3   g[s] ← 0
4   Insertg(OPEN, s)
5   CLOSED ← ∅
6   loop do
7     if IsEmpty(OPEN) then
8       return "failure"
9     v ← DeleteMing(OPEN)
10    CLOSED ← CLOSED ∪ {v}
11    if IsGoal(v) then
12      return Solution(v, s)
13    Expand(v)
14  procedure Expand(v)
15  foreach u ∈ Succ(v) do
16    if u ∉ OPEN ∪ CLOSED then
17      g[u] ← g[v] + c(v, u)
18      Parent[u] ← v
19      Insertg(OPEN, u)
20    else if u ∈ OPEN then
21      if g[v] + c(v, u) < g[u] then
22        g[u] ← g[v] + c(v, u)
23        Parent[u] ← v

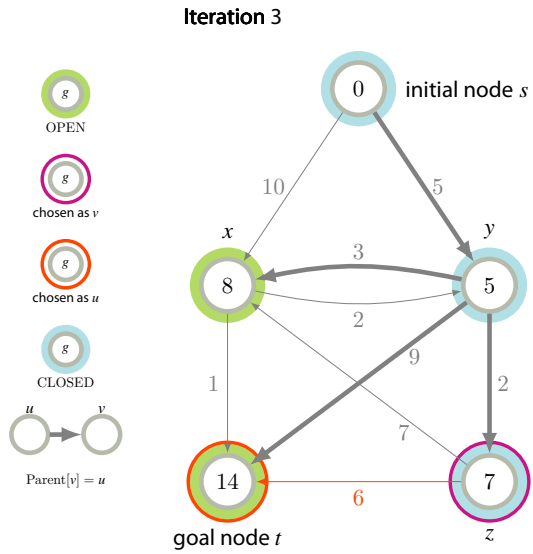
```



```

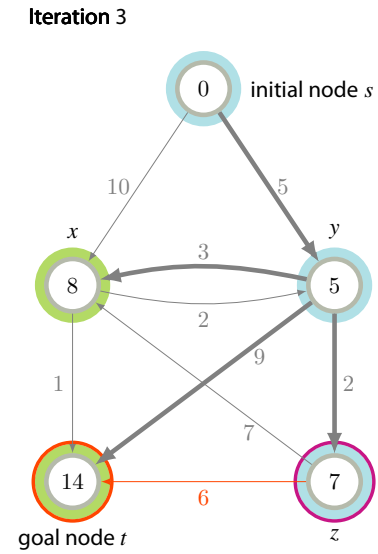
1 function Dijkstra(s)
2   OPEN ← new PriorityQueueg
3   g[s] ← 0
4   Insertg(OPEN, s)
5   CLOSED ← ∅
6   loop do
7     if IsEmpty(OPEN) then
8       return "failure"
9     v ← DeleteMing(OPEN)
10    CLOSED ← CLOSED ∪ {v}
11    if IsGoal(v) then
12      return Solution(v, s)
13    Expand(v)
14  procedure Expand(v)
15  foreach u ∈ Succ(v) do
16    if u ∉ OPEN ∪ CLOSED then
17      g[u] ← g[v] + c(v, u)
18      Parent[u] ← v
19      Insertg(OPEN, u)
20    else if u ∈ OPEN then
21      if g[v] + c(v, u) < g[u] then
22        g[u] ← g[v] + c(v, u)
23        Parent[u] ← v

```



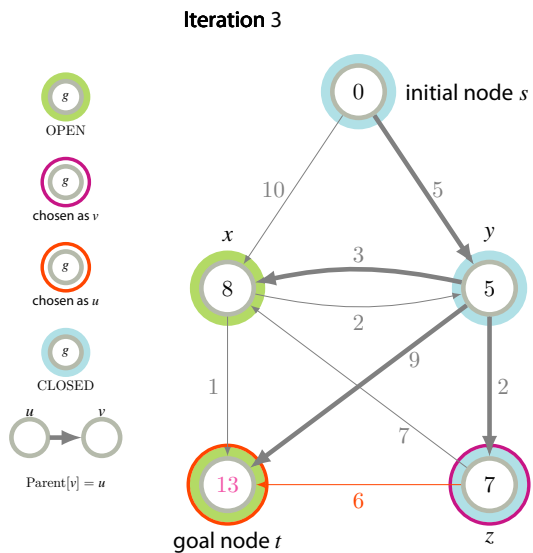
```

1 function Dijkstra(s)
2 OPEN ← new PriorityQueueg
3 g[s] ← 0
4 Insertg(OPEN, s)
5 CLOSED ← ∅
6 loop do
7   if IsEmpty(OPEN) then
8     return "failure"
9   v ← DeleteMing(OPEN)
10  CLOSED ← CLOSED ∪ {v}
11  if IsGoal(v) then
12    return Solution(v, s)
13  Expand(v)
14 procedure Expand(v)
15 foreach u ∈ Succ(v) do
16   if u ∉ OPEN ∪ CLOSED then
17     g[u] ← g[v] + c(v, u)
18     Parent[u] ← v
19     Insertg(OPEN, u)
20   else if u ∈ OPEN then
21     if g[v] + c(v, u) < g[u] then
22       g[u] ← g[v] + c(v, u)
23       Parent[u] ← v
  
```



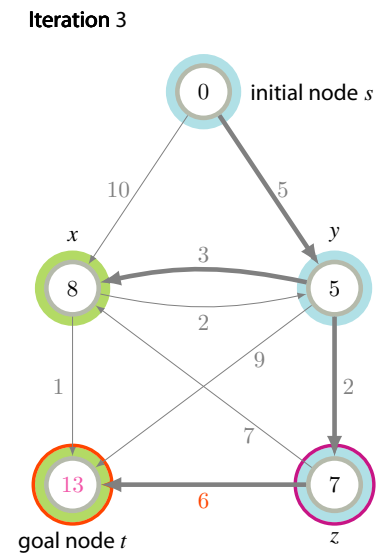
```

1 function Dijkstra(s)
2 OPEN ← new PriorityQueueg
3 g[s] ← 0
4 Insertg(OPEN, s)
5 CLOSED ← ∅
6 loop do
7   if IsEmpty(OPEN) then
8     return "failure"
9   v ← DeleteMing(OPEN)
10  CLOSED ← CLOSED ∪ {v}
11  if IsGoal(v) then
12    return Solution(v, s)
13  Expand(v)
14 procedure Expand(v)
15 foreach u ∈ Succ(v) do
16   if u ∉ OPEN ∪ CLOSED then
17     g[u] ← g[v] + c(v, u)
18     Parent[u] ← v
19     Insertg(OPEN, u)
20   else if u ∈ OPEN then
21     if g[v] + c(v, u) < g[u] then
22       g[u] ← g[v] + c(v, u)
23       Parent[u] ← v
  
```



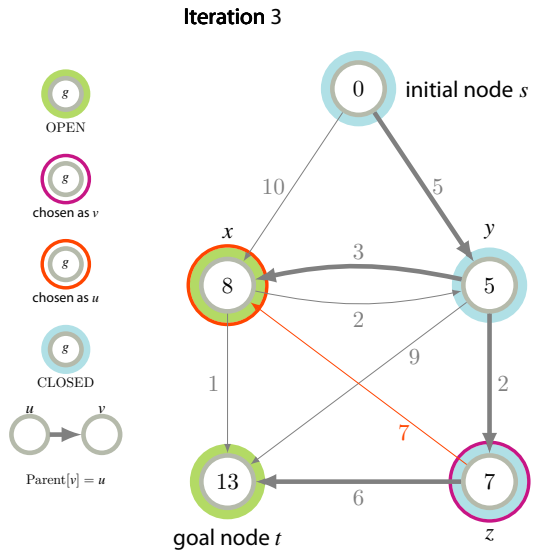
```

1 function Dijkstra(s)
2 OPEN ← new PriorityQueueg
3 g[s] ← 0
4 Insertg(OPEN, s)
5 CLOSED ← ∅
6 loop do
7   if IsEmpty(OPEN) then
8     return "failure"
9   v ← DeleteMing(OPEN)
10  CLOSED ← CLOSED ∪ {v}
11  if IsGoal(v) then
12    return Solution(v, s)
13  Expand(v)
14 procedure Expand(v)
15 foreach u ∈ Succ(v) do
16   if u ∉ OPEN ∪ CLOSED then
17     g[u] ← g[v] + c(v, u)
18     Parent[u] ← v
19     Insertg(OPEN, u)
20   else if u ∈ OPEN then
21     if g[v] + c(v, u) < g[u] then
22       g[u] ← g[v] + c(v, u)
23       Parent[u] ← v
  
```



```

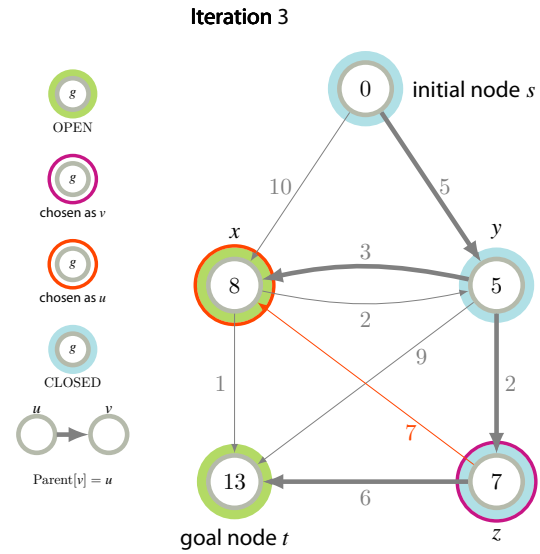
1 function Dijkstra(s)
2 OPEN ← new PriorityQueueg
3 g[s] ← 0
4 Insertg(OPEN, s)
5 CLOSED ← ∅
6 loop do
7   if IsEmpty(OPEN) then
8     return "failure"
9   v ← DeleteMing(OPEN)
10  CLOSED ← CLOSED ∪ {v}
11  if IsGoal(v) then
12    return Solution(v, s)
13  Expand(v)
14 procedure Expand(v)
15 foreach u ∈ Succ(v) do
16   if u ∉ OPEN ∪ CLOSED then
17     g[u] ← g[v] + c(v, u)
18     Parent[u] ← v
19     Insertg(OPEN, u)
20   else if u ∈ OPEN then
21     if g[v] + c(v, u) < g[u] then
22       g[u] ← g[v] + c(v, u)
23       Parent[u] ← v
  
```



```

1 function Dijkstra( $s$ )
2 OPEN  $\leftarrow$  new PriorityQueue $_g$ 
3  $g[s] \leftarrow 0$ 
4 Insert $_g$ (OPEN,  $s$ )
5 CLOSED  $\leftarrow \emptyset$ 
6 loop do
7   if IsEmpty(OPEN) then
8     return "failure"
9    $v \leftarrow$  DeleteMin $_g$ (OPEN)
10  CLOSED  $\leftarrow$  CLOSED  $\cup \{v\}$ 
11  if IsGoal( $v$ ) then
12    return Solution( $v, s$ )
13  Expand( $v$ )
14 procedure Expand( $v$ )
15 foreach  $u \in$  Succ( $v$ ) do
16   if  $u \notin$  OPEN  $\cup$  CLOSED then
17      $g[u] \leftarrow g[v] + c(v, u)$ 
18     Parent[ $u$ ]  $\leftarrow v$ 
19     Insert $_g$ (OPEN,  $u$ )
20   else if  $u \in$  OPEN then
21     if  $g[v] + c(v, u) < g[u]$  then
22        $g[u] \leftarrow g[v] + c(v, u)$ 
23       Parent[ $u$ ]  $\leftarrow v$ 

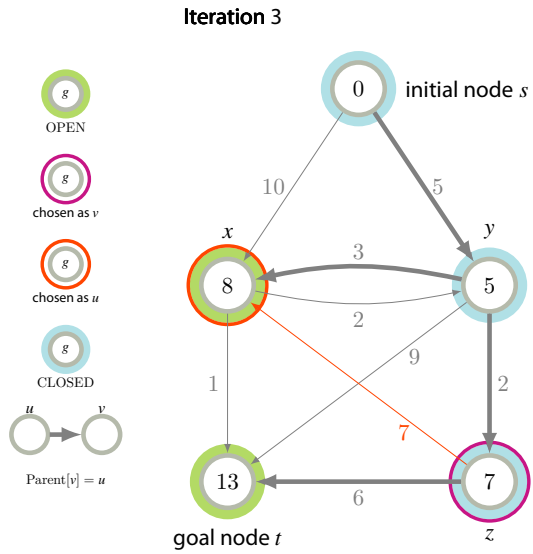
```



```

1 function Dijkstra( $s$ )
2 OPEN  $\leftarrow$  new PriorityQueue $_g$ 
3  $g[s] \leftarrow 0$ 
4 Insert $_g$ (OPEN,  $s$ )
5 CLOSED  $\leftarrow \emptyset$ 
6 loop do
7   if IsEmpty(OPEN) then
8     return "failure"
9    $v \leftarrow$  DeleteMin $_g$ (OPEN)
10  CLOSED  $\leftarrow$  CLOSED  $\cup \{v\}$ 
11  if IsGoal( $v$ ) then
12    return Solution( $v, s$ )
13  Expand( $v$ )
14 procedure Expand( $v$ )
15 foreach  $u \in$  Succ( $v$ ) do
16   if  $u \notin$  OPEN  $\cup$  CLOSED then
17      $g[u] \leftarrow g[v] + c(v, u)$ 
18     Parent[ $u$ ]  $\leftarrow v$ 
19     Insert $_g$ (OPEN,  $u$ )
20   else if  $u \in$  OPEN then
21     if  $g[v] + c(v, u) < g[u]$  then
22        $g[u] \leftarrow g[v] + c(v, u)$ 
23       Parent[ $u$ ]  $\leftarrow v$ 

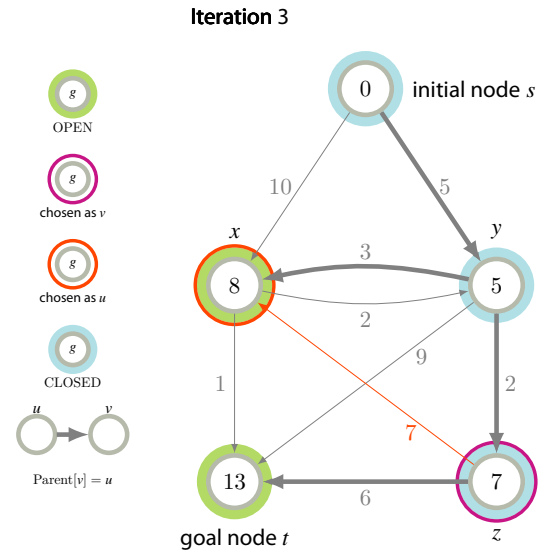
```



```

1 function Dijkstra( $s$ )
2 OPEN  $\leftarrow$  new PriorityQueue $_g$ 
3  $g[s] \leftarrow 0$ 
4 Insert $_g$ (OPEN,  $s$ )
5 CLOSED  $\leftarrow \emptyset$ 
6 loop do
7   if IsEmpty(OPEN) then
8     return "failure"
9    $v \leftarrow$  DeleteMin $_g$ (OPEN)
10  CLOSED  $\leftarrow$  CLOSED  $\cup \{v\}$ 
11  if IsGoal( $v$ ) then
12    return Solution( $v, s$ )
13  Expand( $v$ )
14 procedure Expand( $v$ )
15 foreach  $u \in$  Succ( $v$ ) do
16   if  $u \notin$  OPEN  $\cup$  CLOSED then
17      $g[u] \leftarrow g[v] + c(v, u)$ 
18     Parent[ $u$ ]  $\leftarrow v$ 
19     Insert $_g$ (OPEN,  $u$ )
20   else if  $u \in$  OPEN then
21     if  $g[v] + c(v, u) < g[u]$  then
22        $g[u] \leftarrow g[v] + c(v, u)$ 
23       Parent[ $u$ ]  $\leftarrow v$ 

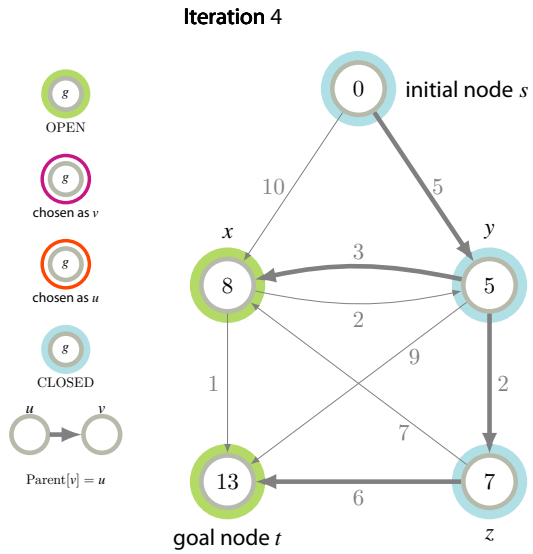
```



```

1 function Dijkstra( $s$ )
2 OPEN  $\leftarrow$  new PriorityQueue $_g$ 
3  $g[s] \leftarrow 0$ 
4 Insert $_g$ (OPEN,  $s$ )
5 CLOSED  $\leftarrow \emptyset$ 
6 loop do
7   if IsEmpty(OPEN) then
8     return "failure"
9    $v \leftarrow$  DeleteMin $_g$ (OPEN)
10  CLOSED  $\leftarrow$  CLOSED  $\cup \{v\}$ 
11  if IsGoal( $v$ ) then
12    return Solution( $v, s$ )
13  Expand( $v$ )
14 procedure Expand( $v$ )
15 foreach  $u \in$  Succ( $v$ ) do
16   if  $u \notin$  OPEN  $\cup$  CLOSED then
17      $g[u] \leftarrow g[v] + c(v, u)$ 
18     Parent[ $u$ ]  $\leftarrow v$ 
19     Insert $_g$ (OPEN,  $u$ )
20   else if  $u \in$  OPEN then
21     if  $g[v] + c(v, u) < g[u]$  then
22        $g[u] \leftarrow g[v] + c(v, u)$ 
23       Parent[ $u$ ]  $\leftarrow v$ 

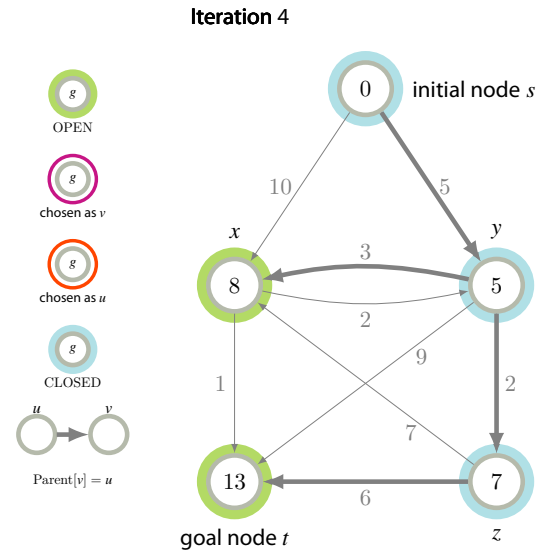
```



```

1 function Dijkstra(s)
2 OPEN ← new PriorityQueueg
3 g[s] ← 0
4 Insertg(OPEN, s)
5 CLOSED ← ∅
6 loop do
7   if IsEmpty(OPEN) then
8     return "failure"
9   v ← DeleteMing(OPEN)
10  CLOSED ← CLOSED ∪ {v}
11  if IsGoal(v) then
12    return Solution(v, s)
13  Expand(v)
14 procedure Expand(v)
15 foreach u ∈ Succ(v) do
16   if u ∉ OPEN ∪ CLOSED then
17     g[u] ← g[v] + c(v, u)
18     Parent[u] ← v
19     Insertg(OPEN, u)
20   else if u ∈ OPEN then
21     if g[v] + c(v, u) < g[u] then
22       g[u] ← g[v] + c(v, u)
23       Parent[u] ← v

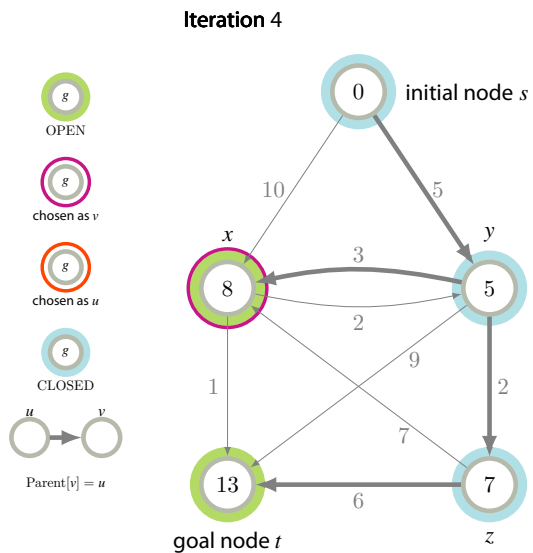
```



```

1 function Dijkstra(s)
2 OPEN ← new PriorityQueueg
3 g[s] ← 0
4 Insertg(OPEN, s)
5 CLOSED ← ∅
6 loop do
7   if IsEmpty(OPEN) then
8     return "failure"
9   v ← DeleteMing(OPEN)
10  CLOSED ← CLOSED ∪ {v}
11  if IsGoal(v) then
12    return Solution(v, s)
13  Expand(v)
14 procedure Expand(v)
15 foreach u ∈ Succ(v) do
16   if u ∉ OPEN ∪ CLOSED then
17     g[u] ← g[v] + c(v, u)
18     Parent[u] ← v
19     Insertg(OPEN, u)
20   else if u ∈ OPEN then
21     if g[v] + c(v, u) < g[u] then
22       g[u] ← g[v] + c(v, u)
23       Parent[u] ← v

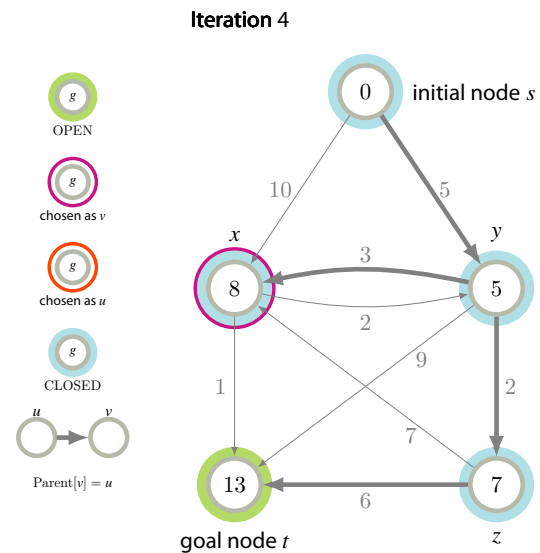
```



```

1 function Dijkstra(s)
2 OPEN ← new PriorityQueueg
3 g[s] ← 0
4 Insertg(OPEN, s)
5 CLOSED ← ∅
6 loop do
7   if IsEmpty(OPEN) then
8     return "failure"
9   v ← DeleteMing(OPEN)
10  CLOSED ← CLOSED ∪ {v}
11  if IsGoal(v) then
12    return Solution(v, s)
13  Expand(v)
14 procedure Expand(v)
15 foreach u ∈ Succ(v) do
16   if u ∉ OPEN ∪ CLOSED then
17     g[u] ← g[v] + c(v, u)
18     Parent[u] ← v
19     Insertg(OPEN, u)
20   else if u ∈ OPEN then
21     if g[v] + c(v, u) < g[u] then
22       g[u] ← g[v] + c(v, u)
23       Parent[u] ← v

```

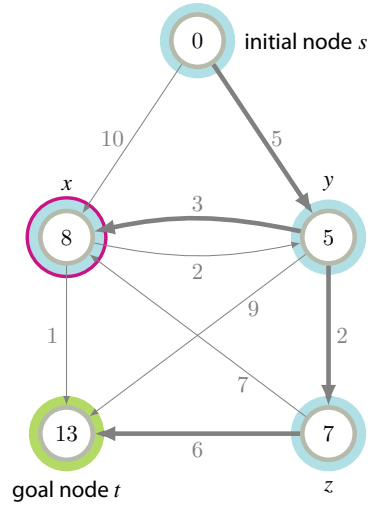


```

1 function Dijkstra(s)
2 OPEN ← new PriorityQueueg
3 g[s] ← 0
4 Insertg(OPEN, s)
5 CLOSED ← ∅
6 loop do
7   if IsEmpty(OPEN) then
8     return "failure"
9   v ← DeleteMing(OPEN)
10  CLOSED ← CLOSED ∪ {v}
11  if IsGoal(v) then
12    return Solution(v, s)
13  Expand(v)
14 procedure Expand(v)
15 foreach u ∈ Succ(v) do
16   if u ∉ OPEN ∪ CLOSED then
17     g[u] ← g[v] + c(v, u)
18     Parent[u] ← v
19     Insertg(OPEN, u)
20   else if u ∈ OPEN then
21     if g[v] + c(v, u) < g[u] then
22       g[u] ← g[v] + c(v, u)
23       Parent[u] ← v

```

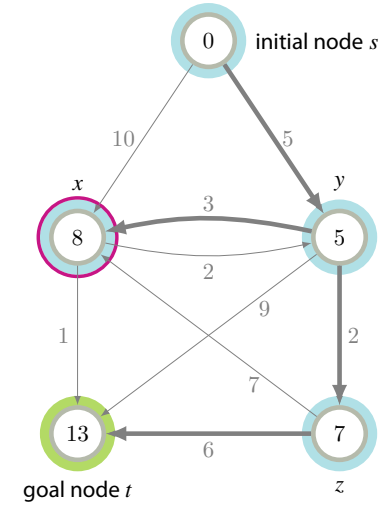

Iteration 4



```

1 function Dijkstra(s)
2 OPEN ← new PriorityQueueg
3 g[s] ← 0
4 Insertg(OPEN, s)
5 CLOSED ← ∅
6 loop do
7   if IsEmpty(OPEN) then
8     return "failure"
9   v ← DeleteMing(OPEN)
10  CLOSED ← CLOSED ∪ {v}
11  if IsGoal(v) then
12    return Solution(v, s)
13  Expand(v)
14 procedure Expand(v)
15 foreach u ∈ Succ(v) do
16   if u ∉ OPEN ∪ CLOSED then
17     g[u] ← g[v] + c(v, u)
18     Parent[u] ← v
19     Insertg(OPEN, u)
20   else if u ∈ OPEN then
21     if g[v] + c(v, u) < g[u] then
22       g[u] ← g[v] + c(v, u)
23       Parent[u] ← v
  
```

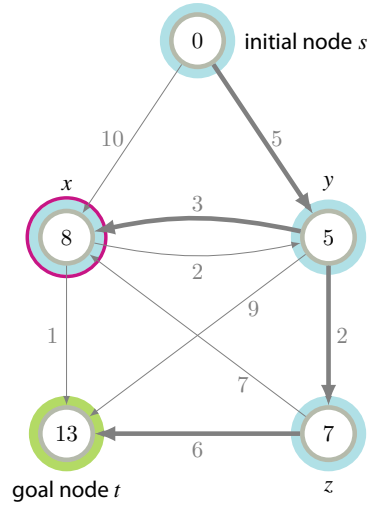
Iteration 4



```

1 function Dijkstra(s)
2 OPEN ← new PriorityQueueg
3 g[s] ← 0
4 Insertg(OPEN, s)
5 CLOSED ← ∅
6 loop do
7   if IsEmpty(OPEN) then
8     return "failure"
9   v ← DeleteMing(OPEN)
10  CLOSED ← CLOSED ∪ {v}
11  if IsGoal(v) then
12    return Solution(v, s)
13  Expand(v)
14 procedure Expand(v)
15 foreach u ∈ Succ(v) do
16   if u ∉ OPEN ∪ CLOSED then
17     g[u] ← g[v] + c(v, u)
18     Parent[u] ← v
19     Insertg(OPEN, u)
20   else if u ∈ OPEN then
21     if g[v] + c(v, u) < g[u] then
22       g[u] ← g[v] + c(v, u)
23       Parent[u] ← v
  
```

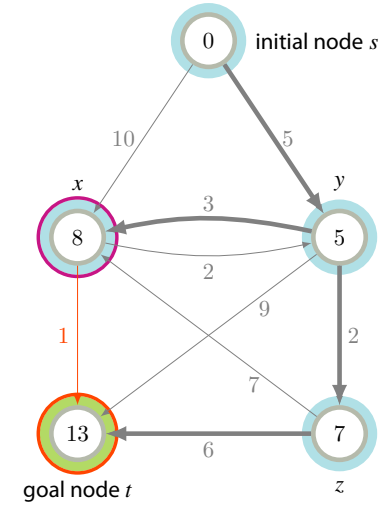
Iteration 4



```

1 function Dijkstra(s)
2 OPEN ← new PriorityQueueg
3 g[s] ← 0
4 Insertg(OPEN, s)
5 CLOSED ← ∅
6 loop do
7   if IsEmpty(OPEN) then
8     return "failure"
9   v ← DeleteMing(OPEN)
10  CLOSED ← CLOSED ∪ {v}
11  if IsGoal(v) then
12    return Solution(v, s)
13  Expand(v)
14 procedure Expand(v)
15 foreach u ∈ Succ(v) do
16   if u ∉ OPEN ∪ CLOSED then
17     g[u] ← g[v] + c(v, u)
18     Parent[u] ← v
19     Insertg(OPEN, u)
20   else if u ∈ OPEN then
21     if g[v] + c(v, u) < g[u] then
22       g[u] ← g[v] + c(v, u)
23       Parent[u] ← v
  
```

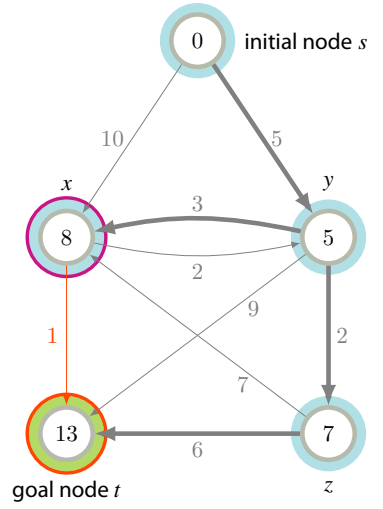
Iteration 4



```

1 function Dijkstra(s)
2 OPEN ← new PriorityQueueg
3 g[s] ← 0
4 Insertg(OPEN, s)
5 CLOSED ← ∅
6 loop do
7   if IsEmpty(OPEN) then
8     return "failure"
9   v ← DeleteMing(OPEN)
10  CLOSED ← CLOSED ∪ {v}
11  if IsGoal(v) then
12    return Solution(v, s)
13  Expand(v)
14 procedure Expand(v)
15 foreach u ∈ Succ(v) do
16   if u ∉ OPEN ∪ CLOSED then
17     g[u] ← g[v] + c(v, u)
18     Parent[u] ← v
19     Insertg(OPEN, u)
20   else if u ∈ OPEN then
21     if g[v] + c(v, u) < g[u] then
22       g[u] ← g[v] + c(v, u)
23       Parent[u] ← v
  
```

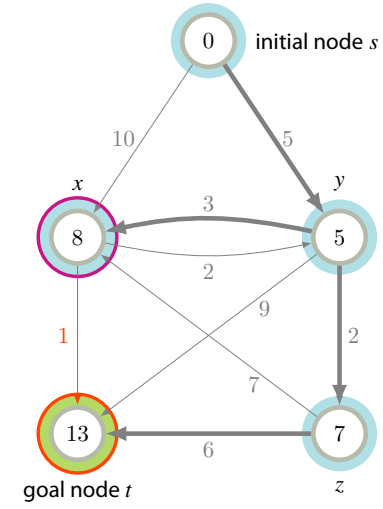
Iteration 4



```

1 function Dijkstra(s)
2   OPEN ← new PriorityQueueg
3   g[s] ← 0
4   Insertg(OPEN, s)
5   CLOSED ← ∅
6   loop do
7     if IsEmpty(OPEN) then
8       return "failure"
9     v ← DeleteMing(OPEN)
10    CLOSED ← CLOSED ∪ {v}
11    if IsGoal(v) then
12      return Solution(v, s)
13    Expand(v)
14  procedure Expand(v)
15  foreach u ∈ Succ(v) do
16    if u ∉ OPEN ∪ CLOSED then
17      g[u] ← g[v] + c(v, u)
18      Parent[u] ← v
19      Insertg(OPEN, u)
20    else if u ∈ OPEN then
21      if g[v] + c(v, u) < g[u] then
22        g[u] ← g[v] + c(v, u)
23        Parent[u] ← v
  
```

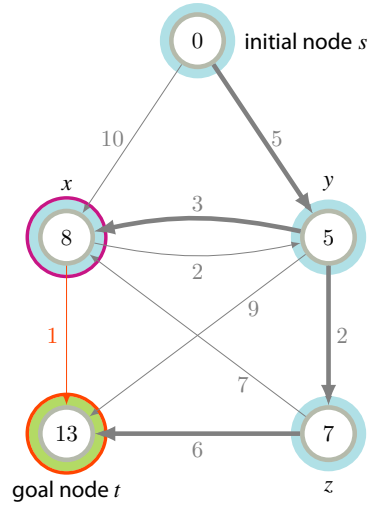
Iteration 4



```

1 function Dijkstra(s)
2   OPEN ← new PriorityQueueg
3   g[s] ← 0
4   Insertg(OPEN, s)
5   CLOSED ← ∅
6   loop do
7     if IsEmpty(OPEN) then
8       return "failure"
9     v ← DeleteMing(OPEN)
10    CLOSED ← CLOSED ∪ {v}
11    if IsGoal(v) then
12      return Solution(v, s)
13    Expand(v)
14  procedure Expand(v)
15  foreach u ∈ Succ(v) do
16    if u ∉ OPEN ∪ CLOSED then
17      g[u] ← g[v] + c(v, u)
18      Parent[u] ← v
19      Insertg(OPEN, u)
20    else if u ∈ OPEN then
21      if g[v] + c(v, u) < g[u] then
22        g[u] ← g[v] + c(v, u)
23        Parent[u] ← v
  
```

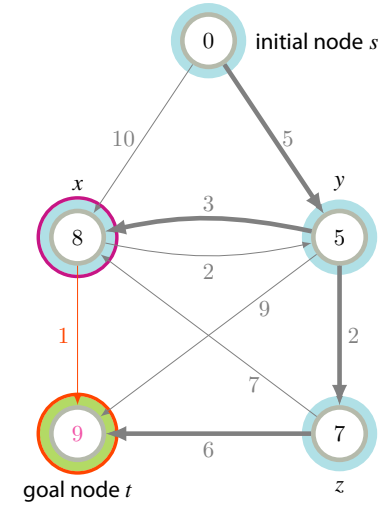
Iteration 4



```

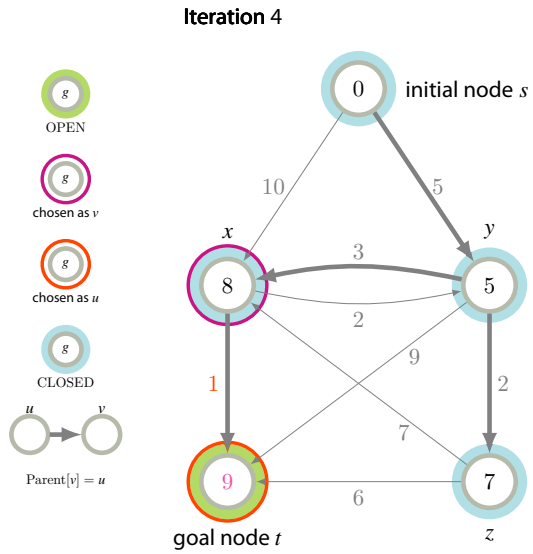
1 function Dijkstra(s)
2   OPEN ← new PriorityQueueg
3   g[s] ← 0
4   Insertg(OPEN, s)
5   CLOSED ← ∅
6   loop do
7     if IsEmpty(OPEN) then
8       return "failure"
9     v ← DeleteMing(OPEN)
10    CLOSED ← CLOSED ∪ {v}
11    if IsGoal(v) then
12      return Solution(v, s)
13    Expand(v)
14  procedure Expand(v)
15  foreach u ∈ Succ(v) do
16    if u ∉ OPEN ∪ CLOSED then
17      g[u] ← g[v] + c(v, u)
18      Parent[u] ← v
19      Insertg(OPEN, u)
20    else if u ∈ OPEN then
21      if g[v] + c(v, u) < g[u] then
22        g[u] ← g[v] + c(v, u)
23        Parent[u] ← v
  
```

Iteration 4



```

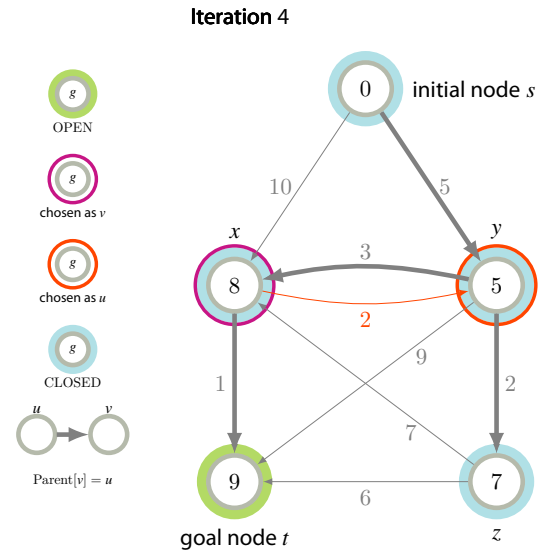
1 function Dijkstra(s)
2   OPEN ← new PriorityQueueg
3   g[s] ← 0
4   Insertg(OPEN, s)
5   CLOSED ← ∅
6   loop do
7     if IsEmpty(OPEN) then
8       return "failure"
9     v ← DeleteMing(OPEN)
10    CLOSED ← CLOSED ∪ {v}
11    if IsGoal(v) then
12      return Solution(v, s)
13    Expand(v)
14  procedure Expand(v)
15  foreach u ∈ Succ(v) do
16    if u ∉ OPEN ∪ CLOSED then
17      g[u] ← g[v] + c(v, u)
18      Parent[u] ← v
19      Insertg(OPEN, u)
20    else if u ∈ OPEN then
21      if g[v] + c(v, u) < g[u] then
22        g[u] ← g[v] + c(v, u)
23        Parent[u] ← v
  
```



```

1 function Dijkstra(s)
2   OPEN ← new PriorityQueueg
3   g[s] ← 0
4   Insertg(OPEN, s)
5   CLOSED ← ∅
6   loop do
7     if IsEmpty(OPEN) then
8       return "failure"
9     v ← DeleteMing(OPEN)
10    CLOSED ← CLOSED ∪ {v}
11    if IsGoal(v) then
12      return Solution(v, s)
13    Expand(v)
14  procedure Expand(v)
15  foreach u ∈ Succ(v) do
16    if u ∉ OPEN ∪ CLOSED then
17      g[u] ← g[v] + c(v, u)
18      Parent[u] ← v
19      Insertg(OPEN, u)
20    else if u ∈ OPEN then
21      if g[v] + c(v, u) < g[u] then
22        g[u] ← g[v] + c(v, u)
23        Parent[u] ← v

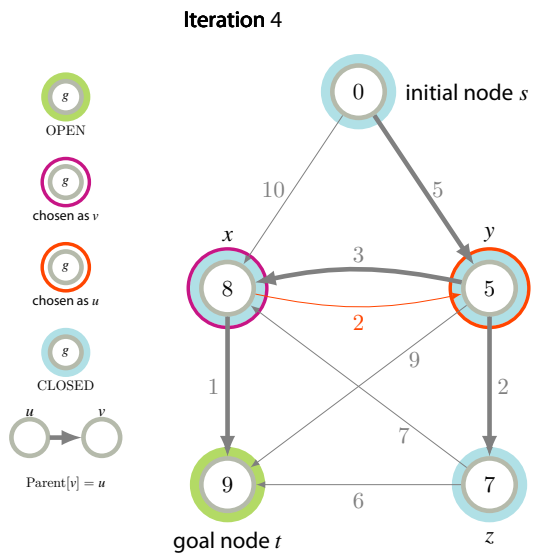
```



```

1 function Dijkstra(s)
2   OPEN ← new PriorityQueueg
3   g[s] ← 0
4   Insertg(OPEN, s)
5   CLOSED ← ∅
6   loop do
7     if IsEmpty(OPEN) then
8       return "failure"
9     v ← DeleteMing(OPEN)
10    CLOSED ← CLOSED ∪ {v}
11    if IsGoal(v) then
12      return Solution(v, s)
13    Expand(v)
14  procedure Expand(v)
15  foreach u ∈ Succ(v) do
16    if u ∉ OPEN ∪ CLOSED then
17      g[u] ← g[v] + c(v, u)
18      Parent[u] ← v
19      Insertg(OPEN, u)
20    else if u ∈ OPEN then
21      if g[v] + c(v, u) < g[u] then
22        g[u] ← g[v] + c(v, u)
23        Parent[u] ← v

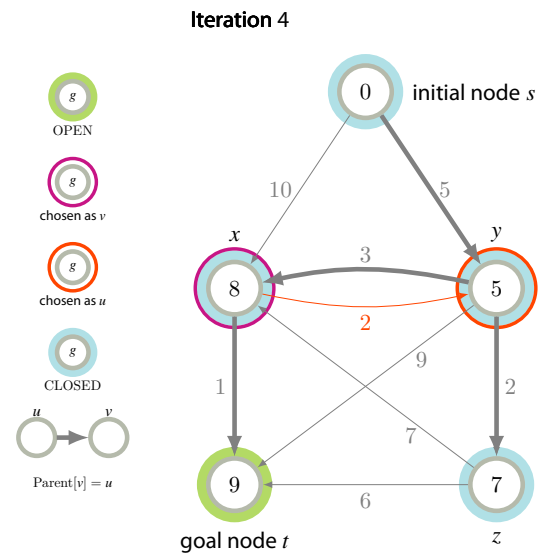
```



```

1 function Dijkstra(s)
2   OPEN ← new PriorityQueueg
3   g[s] ← 0
4   Insertg(OPEN, s)
5   CLOSED ← ∅
6   loop do
7     if IsEmpty(OPEN) then
8       return "failure"
9     v ← DeleteMing(OPEN)
10    CLOSED ← CLOSED ∪ {v}
11    if IsGoal(v) then
12      return Solution(v, s)
13    Expand(v)
14  procedure Expand(v)
15  foreach u ∈ Succ(v) do
16    if u ∉ OPEN ∪ CLOSED then
17      g[u] ← g[v] + c(v, u)
18      Parent[u] ← v
19      Insertg(OPEN, u)
20    else if u ∈ OPEN then
21      if g[v] + c(v, u) < g[u] then
22        g[u] ← g[v] + c(v, u)
23        Parent[u] ← v

```

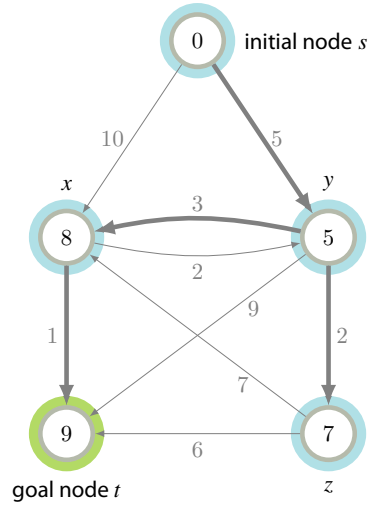


```

1 function Dijkstra(s)
2   OPEN ← new PriorityQueueg
3   g[s] ← 0
4   Insertg(OPEN, s)
5   CLOSED ← ∅
6   loop do
7     if IsEmpty(OPEN) then
8       return "failure"
9     v ← DeleteMing(OPEN)
10    CLOSED ← CLOSED ∪ {v}
11    if IsGoal(v) then
12      return Solution(v, s)
13    Expand(v)
14  procedure Expand(v)
15  foreach u ∈ Succ(v) do
16    if u ∉ OPEN ∪ CLOSED then
17      g[u] ← g[v] + c(v, u)
18      Parent[u] ← v
19      Insertg(OPEN, u)
20    else if u ∈ OPEN then
21      if g[v] + c(v, u) < g[u] then
22        g[u] ← g[v] + c(v, u)
23        Parent[u] ← v

```

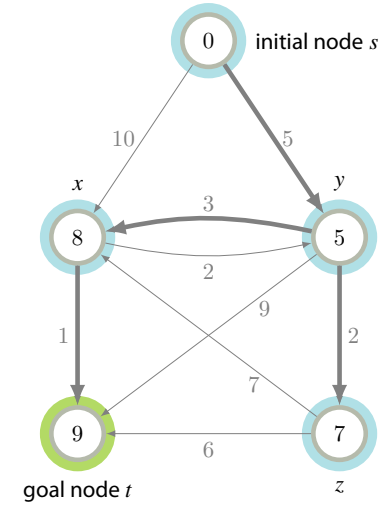
Iteration 5



```

1 function Dijkstra(s)
2 OPEN ← new PriorityQueueg
3 g[s] ← 0
4 Insertg(OPEN, s)
5 CLOSED ← ∅
6 loop do
7   if IsEmpty(OPEN) then
8     return "failure"
9   v ← DeleteMing(OPEN)
10  CLOSED ← CLOSED ∪ {v}
11  if IsGoal(v) then
12    return Solution(v, s)
13  Expand(v)
14 procedure Expand(v)
15 foreach u ∈ Succ(v) do
16   if u ∉ OPEN ∪ CLOSED then
17     g[u] ← g[v] + c(v, u)
18     Parent[u] ← v
19     Insertg(OPEN, u)
20   else if u ∈ OPEN then
21     if g[v] + c(v, u) < g[u] then
22       g[u] ← g[v] + c(v, u)
23       Parent[u] ← v
  
```

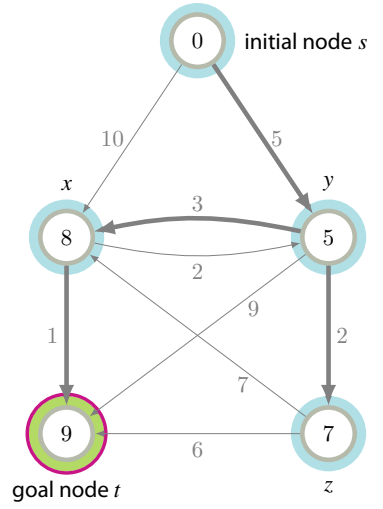
Iteration 5



```

1 function Dijkstra(s)
2 OPEN ← new PriorityQueueg
3 g[s] ← 0
4 Insertg(OPEN, s)
5 CLOSED ← ∅
6 loop do
7   if IsEmpty(OPEN) then
8     return "failure"
9   v ← DeleteMing(OPEN)
10  CLOSED ← CLOSED ∪ {v}
11  if IsGoal(v) then
12    return Solution(v, s)
13  Expand(v)
14 procedure Expand(v)
15 foreach u ∈ Succ(v) do
16   if u ∉ OPEN ∪ CLOSED then
17     g[u] ← g[v] + c(v, u)
18     Parent[u] ← v
19     Insertg(OPEN, u)
20   else if u ∈ OPEN then
21     if g[v] + c(v, u) < g[u] then
22       g[u] ← g[v] + c(v, u)
23       Parent[u] ← v
  
```

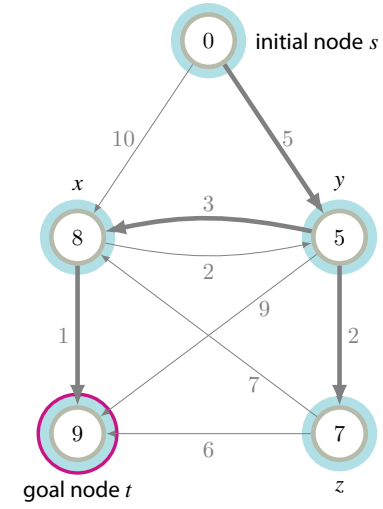
Iteration 5



```

1 function Dijkstra(s)
2 OPEN ← new PriorityQueueg
3 g[s] ← 0
4 Insertg(OPEN, s)
5 CLOSED ← ∅
6 loop do
7   if IsEmpty(OPEN) then
8     return "failure"
9   v ← DeleteMing(OPEN)
10  CLOSED ← CLOSED ∪ {v}
11  if IsGoal(v) then
12    return Solution(v, s)
13  Expand(v)
14 procedure Expand(v)
15 foreach u ∈ Succ(v) do
16   if u ∉ OPEN ∪ CLOSED then
17     g[u] ← g[v] + c(v, u)
18     Parent[u] ← v
19     Insertg(OPEN, u)
20   else if u ∈ OPEN then
21     if g[v] + c(v, u) < g[u] then
22       g[u] ← g[v] + c(v, u)
23       Parent[u] ← v
  
```

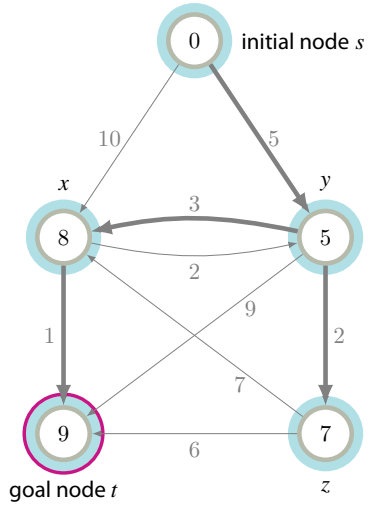
Iteration 5



```

1 function Dijkstra(s)
2 OPEN ← new PriorityQueueg
3 g[s] ← 0
4 Insertg(OPEN, s)
5 CLOSED ← ∅
6 loop do
7   if IsEmpty(OPEN) then
8     return "failure"
9   v ← DeleteMing(OPEN)
10  CLOSED ← CLOSED ∪ {v}
11  if IsGoal(v) then
12    return Solution(v, s)
13  Expand(v)
14 procedure Expand(v)
15 foreach u ∈ Succ(v) do
16   if u ∉ OPEN ∪ CLOSED then
17     g[u] ← g[v] + c(v, u)
18     Parent[u] ← v
19     Insertg(OPEN, u)
20   else if u ∈ OPEN then
21     if g[v] + c(v, u) < g[u] then
22       g[u] ← g[v] + c(v, u)
23       Parent[u] ← v
  
```

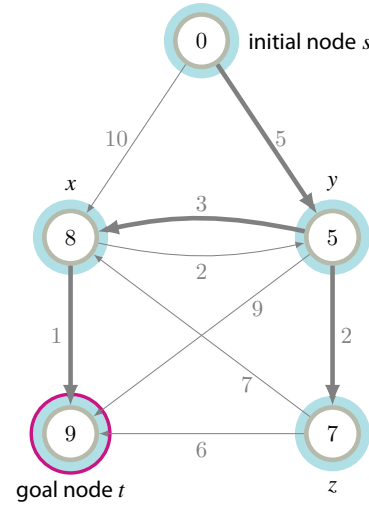
Iteration 5



```

1 function Dijkstra(s)
2   OPEN ← new PriorityQueueg
3   g[s] ← 0
4   Insertg(OPEN, s)
5   CLOSED ← ∅
6   loop do
7     if IsEmpty(OPEN) then
8       return "failure"
9     v ← DeleteMing(OPEN)
10    CLOSED ← CLOSED ∪ {v}
11    if IsGoal(v) then
12      return Solution(v, s)
13    Expand(v)
14  procedure Expand(v)
15  foreach u ∈ Succ(v) do
16    if u ∉ OPEN ∪ CLOSED then
17      g[u] ← g[v] + c(v, u)
18      Parent[u] ← v
19      Insertg(OPEN, u)
20    else if u ∈ OPEN then
21      if g[v] + c(v, u) < g[u] then
22        g[u] ← g[v] + c(v, u)
23        Parent[u] ← v
  
```

Iteration 5



```

1 function Dijkstra(s)
2   OPEN ← new PriorityQueueg
3   g[s] ← 0
4   Insertg(OPEN, s)
5   CLOSED ← ∅
6   loop do
7     if IsEmpty(OPEN) then
8       return "failure"
9     v ← DeleteMing(OPEN)
10    CLOSED ← CLOSED ∪ {v}
11    if IsGoal(v) then
12      return Solution(v, s)
13    Expand(v)
14  procedure Expand(v)
15  foreach u ∈ Succ(v) do
16    if u ∉ OPEN ∪ CLOSED then
17      g[u] ← g[v] + c(v, u)
18      Parent[u] ← v
19      Insertg(OPEN, u)
20    else if u ∈ OPEN then
21      if g[v] + c(v, u) < g[u] then
22        g[u] ← g[v] + c(v, u)
23        Parent[u] ← v
  
```