The Dijkstra Algorithm

Antonio Carzaniga

Faculty of Informatics
Università della Svizzera italiana

May 15, 2018

Dijkstra's Algorithm

- Executing locally at node u

Dijkstra's Algorithm

- Executing locally at node u
- Variables storing values known at each iteration

Dijkstra's Algorithm

- Executing locally at node u
- Variables storing values known at each iteration
  - $D[v]$, cost of the least-cost path from u to v

Dijkstra's Algorithm

- Executing locally at node u
- Variables storing values known at each iteration
  - $D[v]$, cost of the least-cost path from u to v
  - $p[v]$, node preceding v (neighbor of v) on the least-cost path from u to v

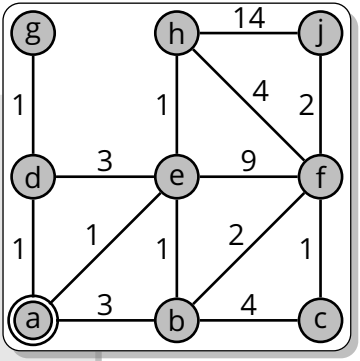
Dijkstra's Algorithm

- Executing locally at node u
- Variables storing values known at each iteration
  - $D[v]$, cost of the least-cost path from u to v
  - $p[v]$, node preceding v (neighbor of v) on the least-cost path from u to v
  - N, nodes of G whose least-cost path from u is definitely known

Dijkstra's Algorithm

```
DIJKSTRA(G = (V, E), u)
N = {u}
for all v ∈ V
    if v ∈ neighbors(u)
        D[v] = c(u, v)
        p[v] = u
    else D[v] = ∞
while N ≠ V
    find w ∉ N such that D[w] is minimum
    N = N ∪ {w}
    for all v ∈ neighbors(w) \ N
        if D[w] + c(w, v) < D[v]
            D[v] = D[w] + c(w, v)
            p[v] = w
```

Example




```
DIJKSTRA(G = (V, E), u)
N = {u}
for all v ∈ V
    if v ∈ neighbors(u)
        D[v] = c(u, v)
        p[v] = u
    else D[v] = ∞
while N ≠ V
    find w ∉ N such that D[w] is minimum
    N = N ∪ {w}
    for all v ∈ neighbors(w) \ N
        if D[w] + c(w, v) < D[v]
            D[v] = D[w] + c(w, v)
            p[v] = w
```