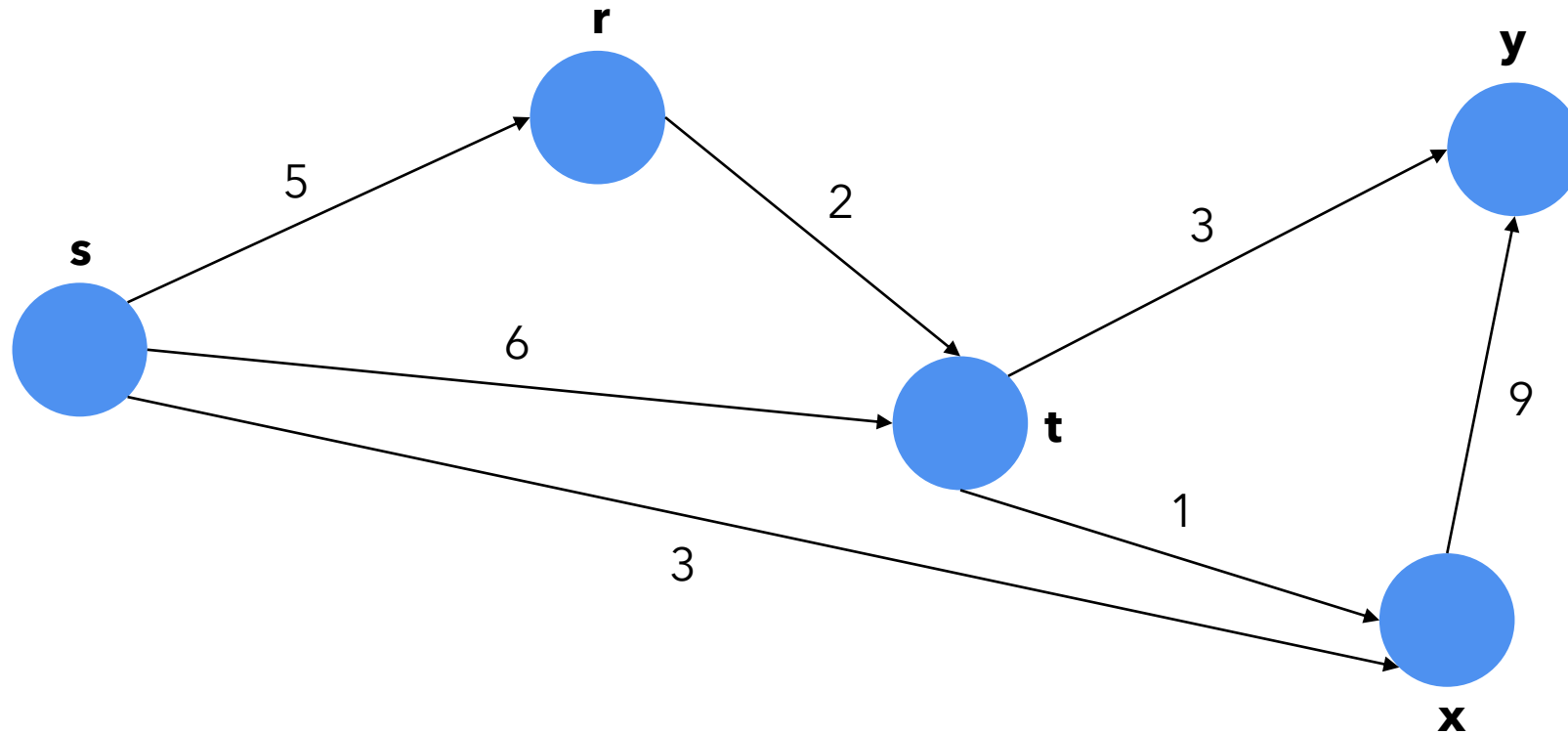


L'algoritmo di Dijkstra ***(single-source shortest-path algorithm)***

Liceo G.B. Brocchi – Bassano del Grappa (VI)
Liceo Scientifico – opzione scienze applicate
Giovanni Mazzocchin

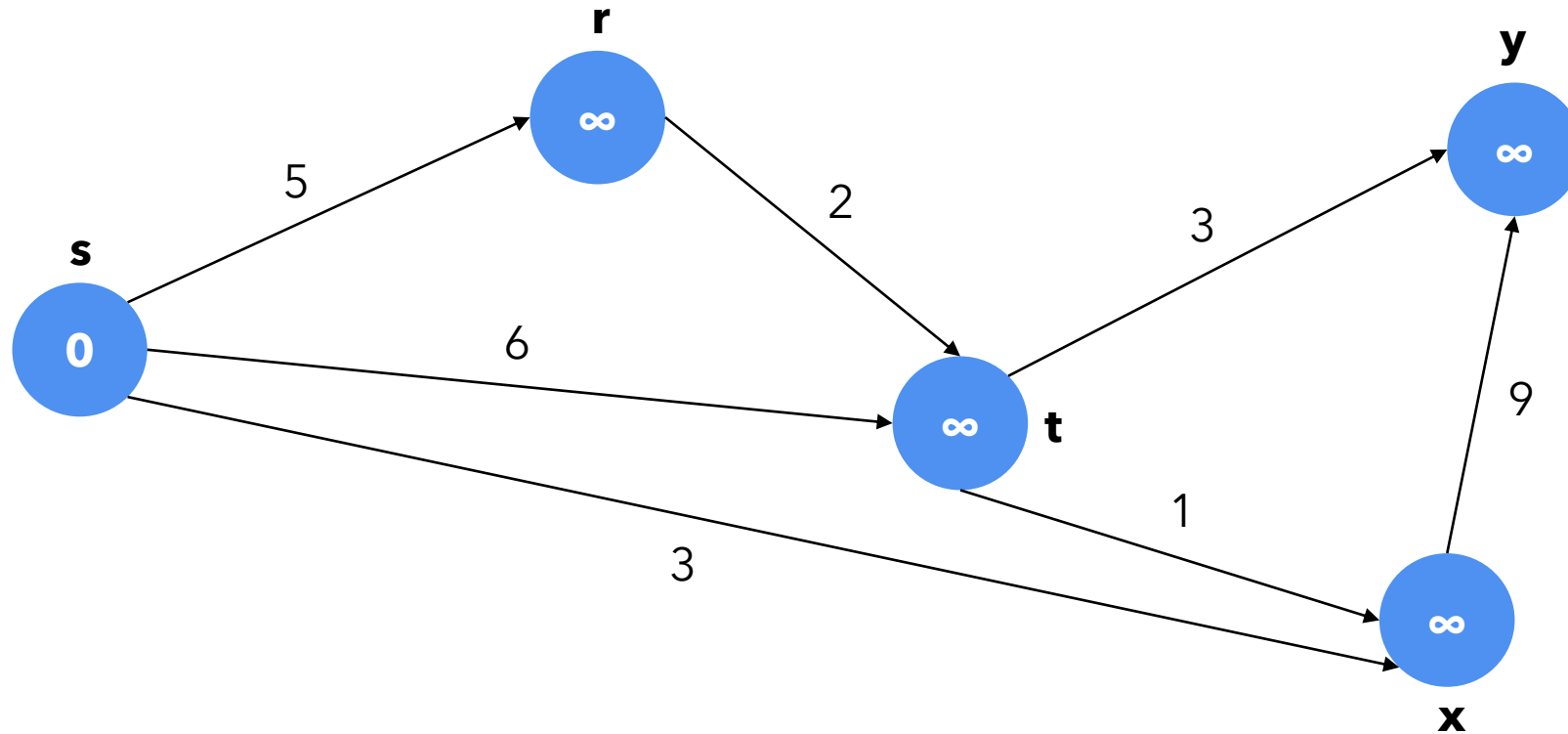
Grafi e cammini minimi (*shortest paths*)



l'obiettivo è calcolare i cammini minimi da s (nodo sorgente) verso tutti gli altri nodi del grafo

- **s, r, t, x, y: nodi** (*vertices*) del grafo
- **s→r, s→t, s→x, r→t, t→x, t→y, x→y:** archi (*edges*) del grafo
- ad ogni arco è associato un **costo**

Grafi e cammini minimi



- ciascun nodo v ha come chiave il costo noto all'iterazione corrente del cammino minimo $s \rightarrow v$
- prima di eseguire l'algoritmo, il costo viene considerato infinito per tutti i nodi, escluso il nodo sorgente

L'algoritmo di Dijkstra

Initialization steps:

1. Consider a directed, weighted graph $G = (V, E)$, where V is the set of vertices, E is the set of edges;
2. Select a source node, name it s . Shortest paths will be computed with s as source;
3. Each v in V will contain the minimum cost of $s \rightarrow v$ computed so far ($+\infty$ at the beginning);
4. Create a set called Q and put $Q = V$ (i.e. assign the set of vertices to Q);

N.B.: costs must be all ≥ 0

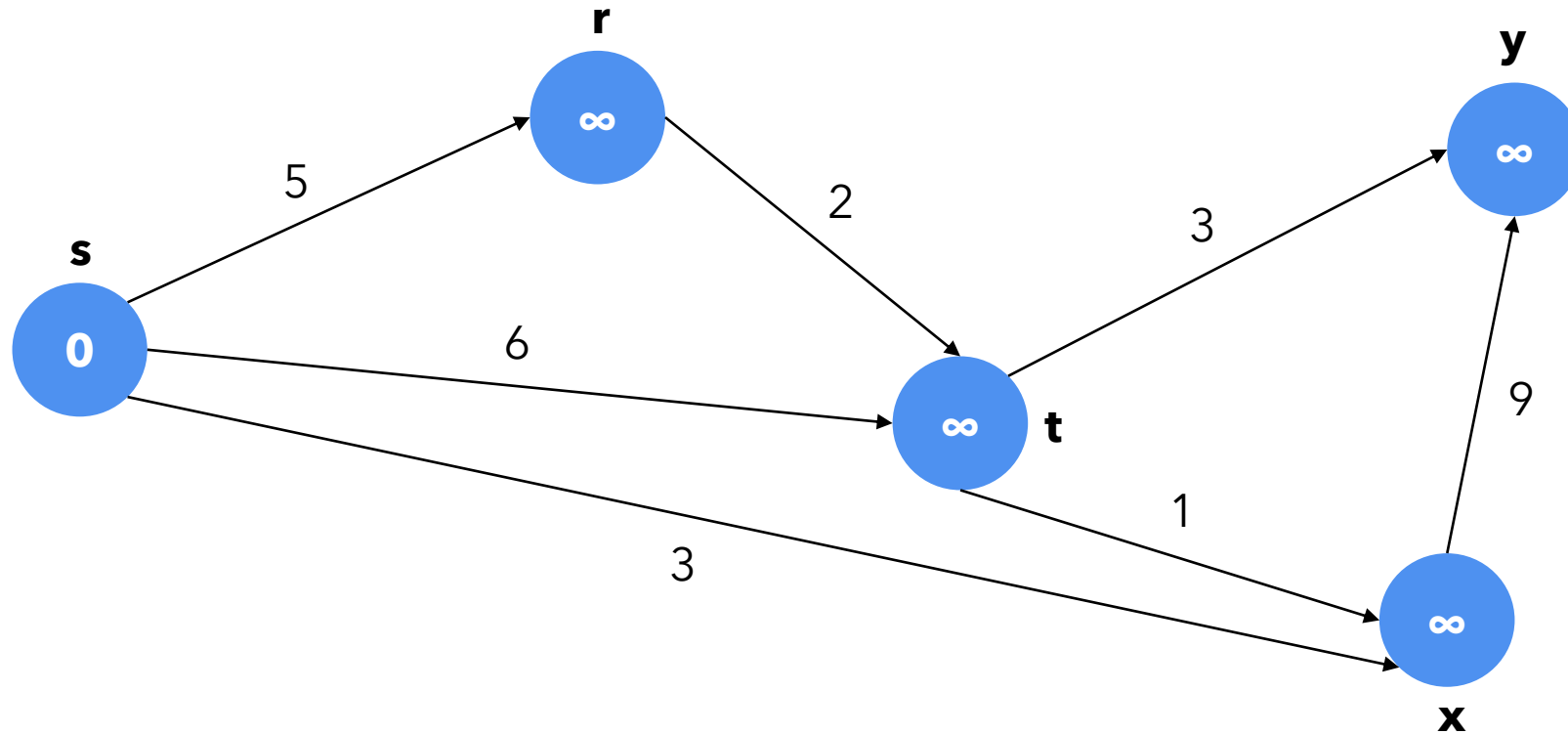
L'algoritmo di Dijkstra

Procedure:

As long as Q is not empty:

1. extract from Q the node with minimum cost, name it v
2. for each u , successor of v :
 - if $v.\text{cost} + \text{cost}(v, u) < u.\text{cost}$:
 - $u.\text{cost} = v.\text{cost} + \text{cost}(v, u)$
 - $u.\text{predecessor} = v$
 - update Q accordingly

L'algoritmo di Dijkstra

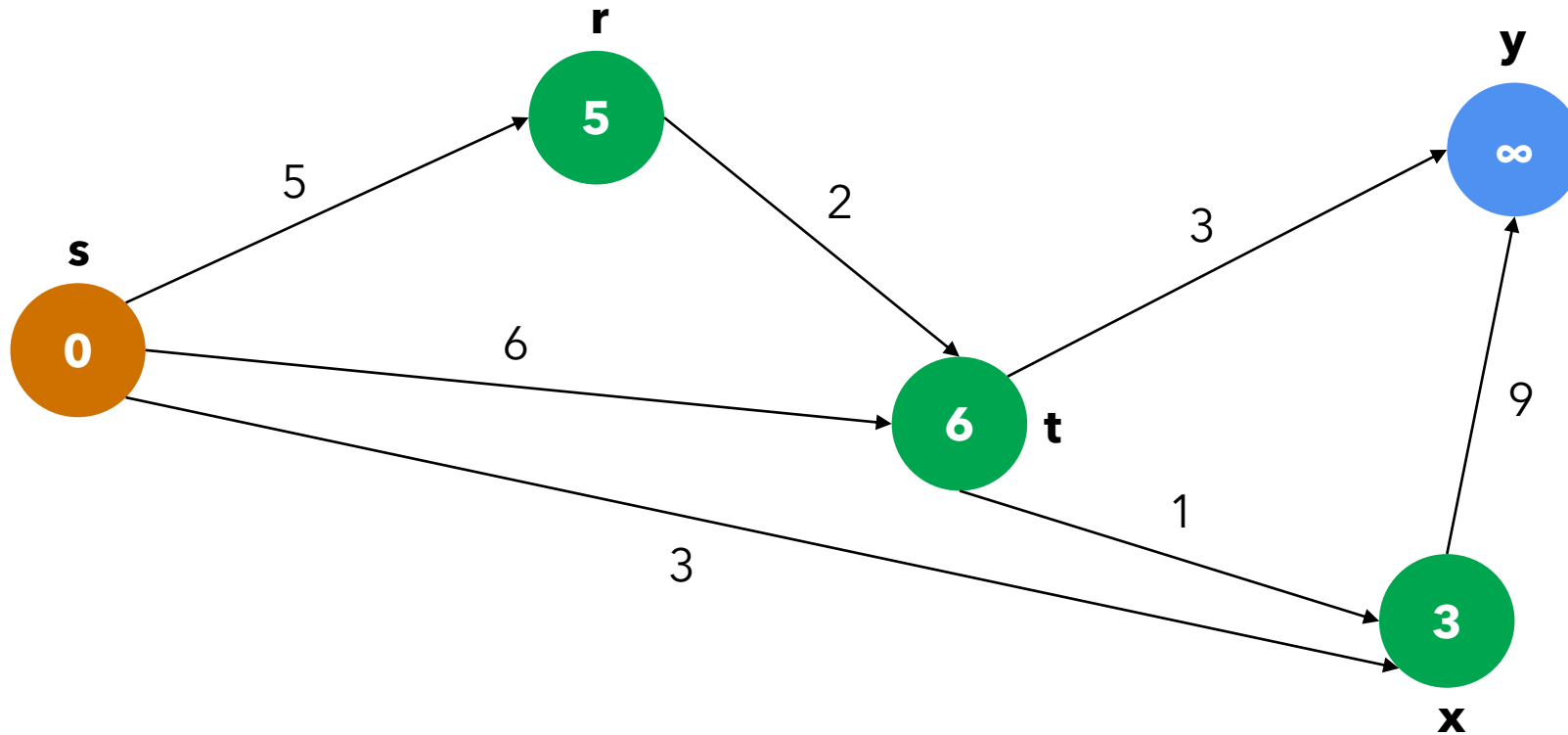


Q: $(s,0)$, (r,inf) , (t,inf) , (x,inf) , (y,inf)

**vertice da cui
inizia la visita ai
successori**

**vertici visitati ed
eventualmente
aggiornati**

L'algoritmo di Dijkstra

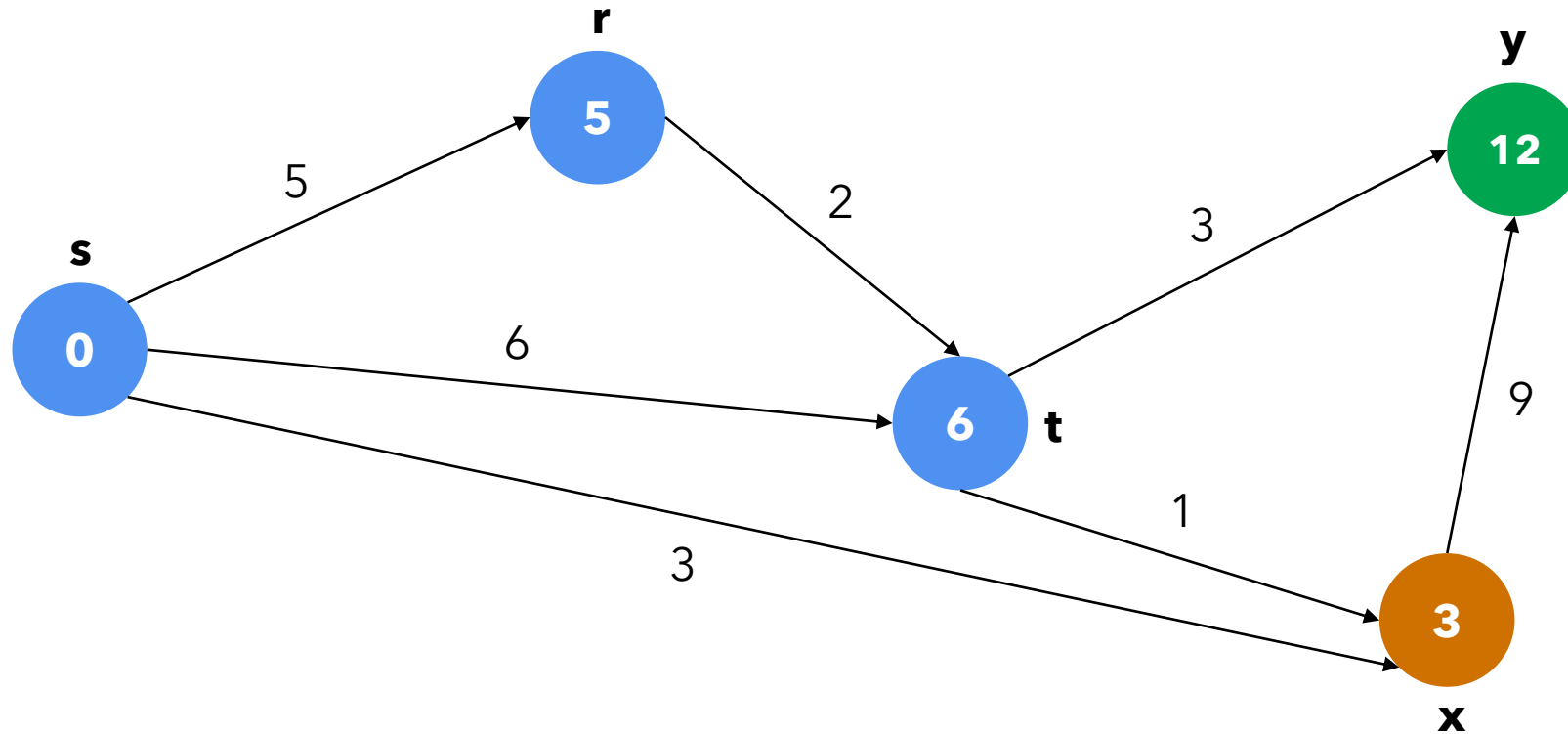


**vertice da cui
inizia la visita ai
successori**

**vertici visitati ed
eventualmente
aggiornati**

Q: (r,5), (t,6), (x,3), (y,inf); extracted node: s
r.predecessor = s
t.predecessor = s
x.predecessor = s

L'algoritmo di Dijkstra

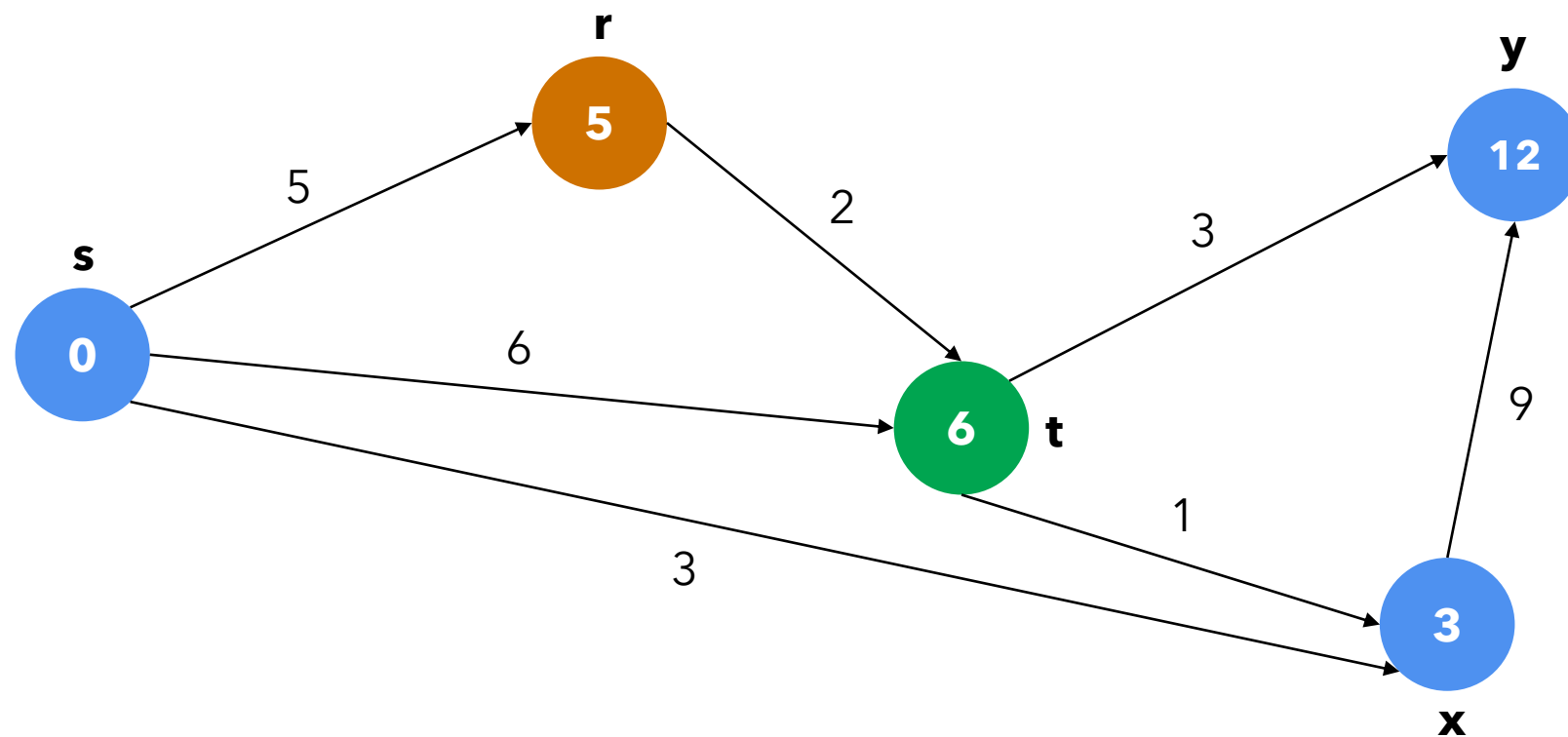


Q: (r,5), (t,6), (y,12); extracted node: x
y.predecessor = x

**vertice da cui
inizia la visita ai
successori**

**vertici visitati ed
eventualmente
aggiornati**

L'algoritmo di Dijkstra

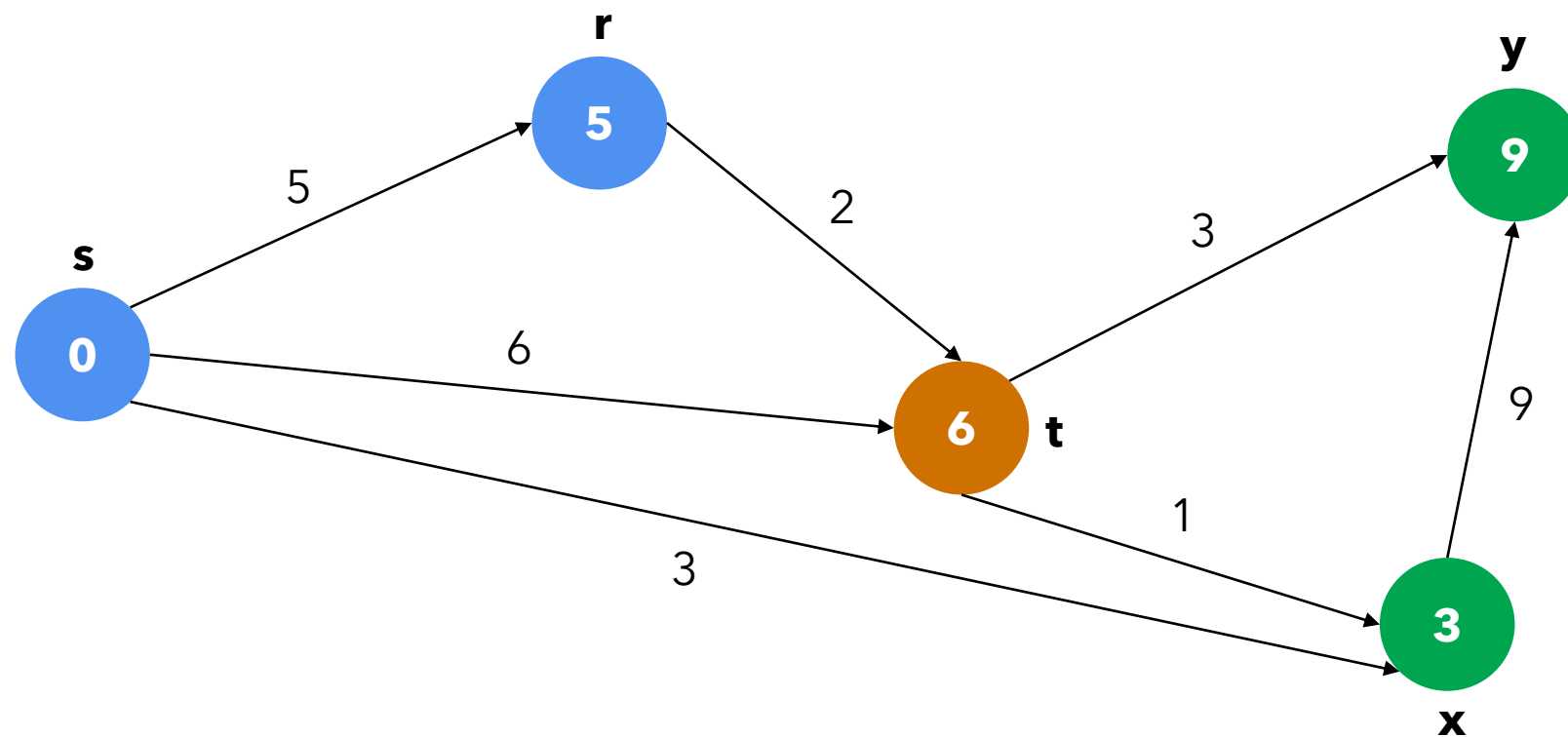


Q: (t,6), (y,12); extracted node: r

**vertice da cui
inizia la visita ai
successori**

**vertici visitati ed
eventualmente
aggiornati**

L'algoritmo di Dijkstra

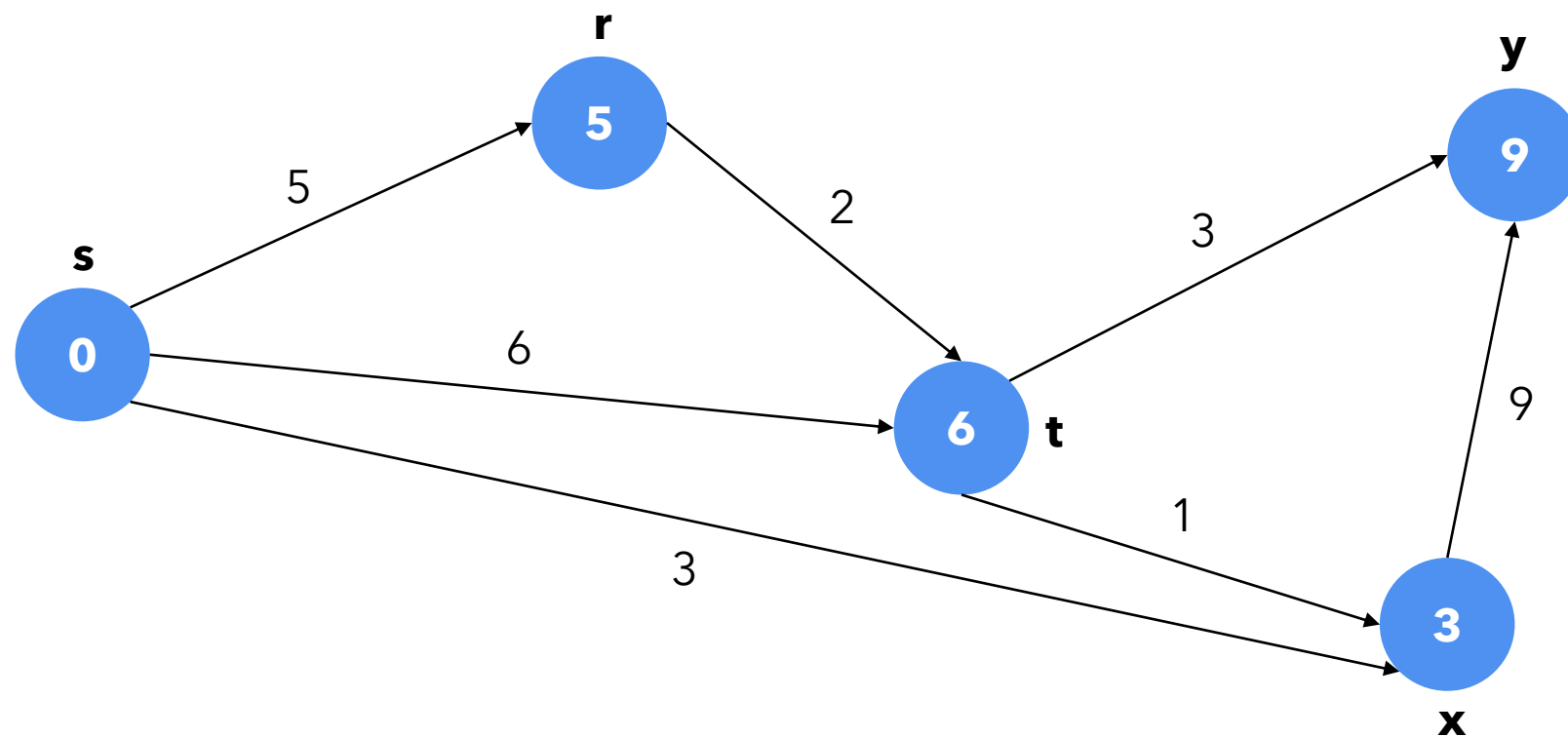


Q: (y,9); extracted node: t
y.predecessor = t

**vertice da cui
inizia la visita ai
successori**

**vertici visitati ed
eventualmente
aggiornati**

L'algoritmo di Dijkstra



Q: empty

**vertice da cui
inizia la visita ai
successori**

**vertici visitati ed
eventualmente
aggiornati**