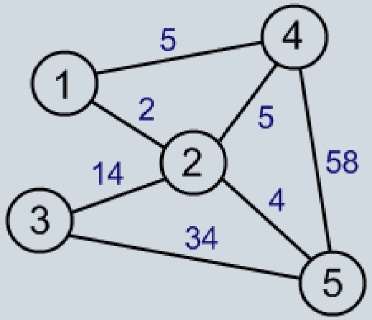


Rappel / démonstration des plus courts chemins avec l'algorithme de *Dijkstra (1959)*

Voir https://algs4.cs.princeton.edu/lectures/keynote/44ShortestPaths.pdf

Plus courts chemins (une source unique, toutes les destinations)

Antécédent: $G = (V, E)$, un graphe connexe avec V l'ensemble des sommets et E l'ensemble des arêtes de G

Pour tout $e \in E$, on note $l(e)$ la longueur de l'arête e

On choisit un sommet source $s \in V$

Conséquent: pour tous les sommets u atteignables depuis s, $dist(u)$ est la distance de s à u

Initialisation:

$\forall\, u \in V$: $dist(u) = \infty$; $prev(u) = nil$

$dist(s) = 0$

Procédure:

$H =$ PriorityQueue(V) // compareTo dist()

while !$H.isEmpty$() **:**

  $u = \mathbf{poll}(H)$

  foreach $e = \{u, v\} \in E$ **:**

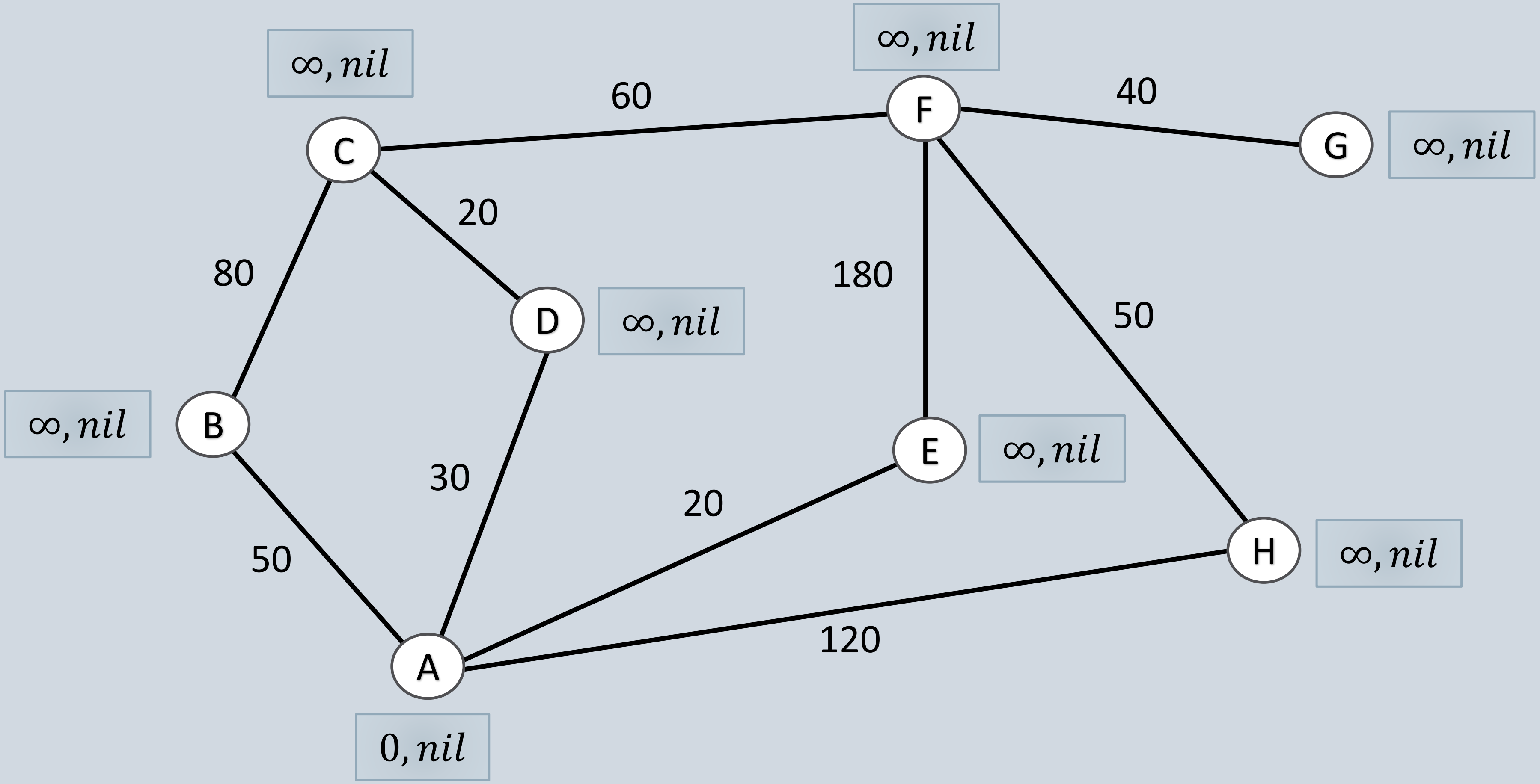
    if $dist(v) > dist(u) + l(u, v)$ **:**

      $dist(v) = dist(u) + l(u, v)$

      $prev(u) = u$

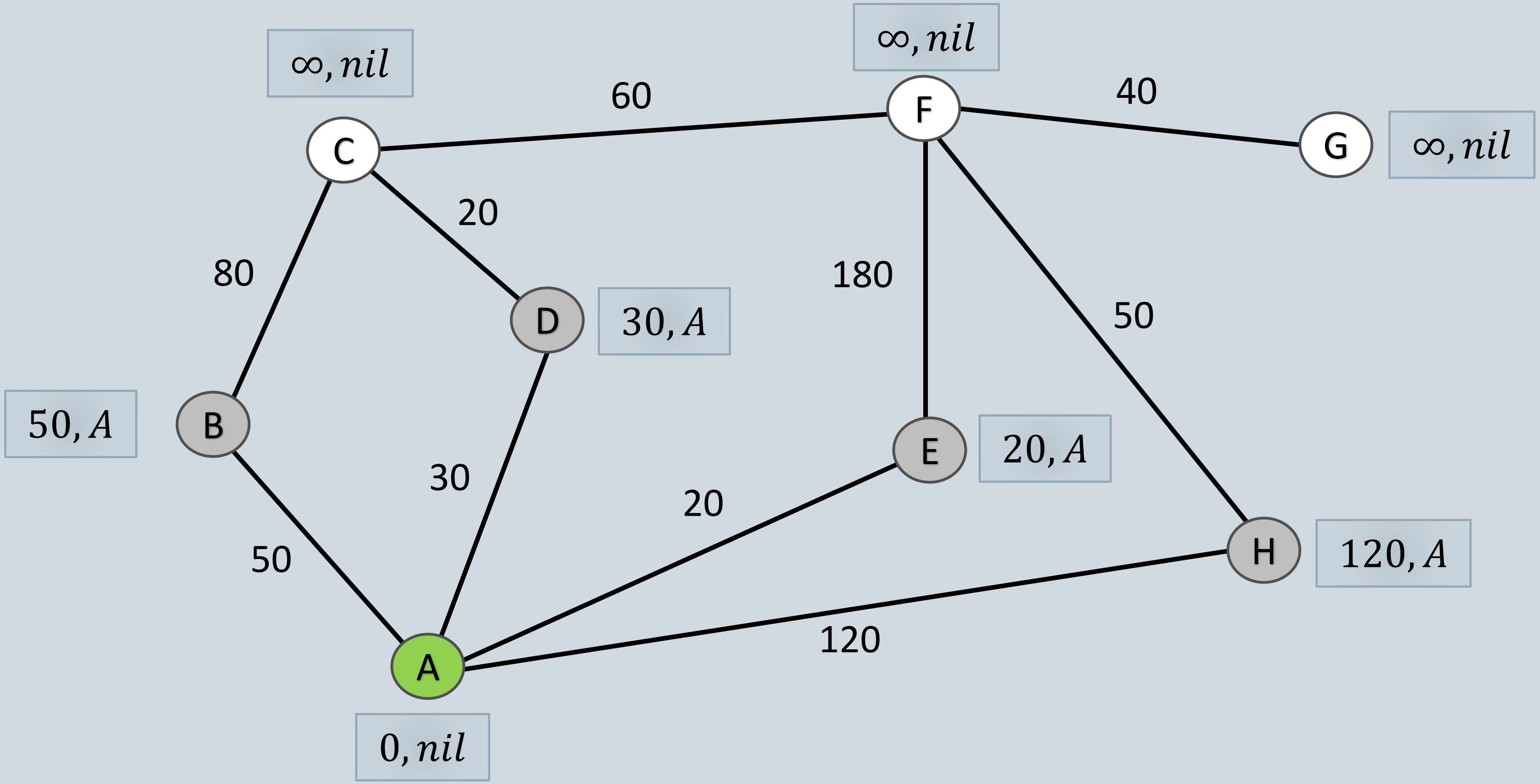
      $\mathbf{update}(H, v)$ // remove puis add

Exemple



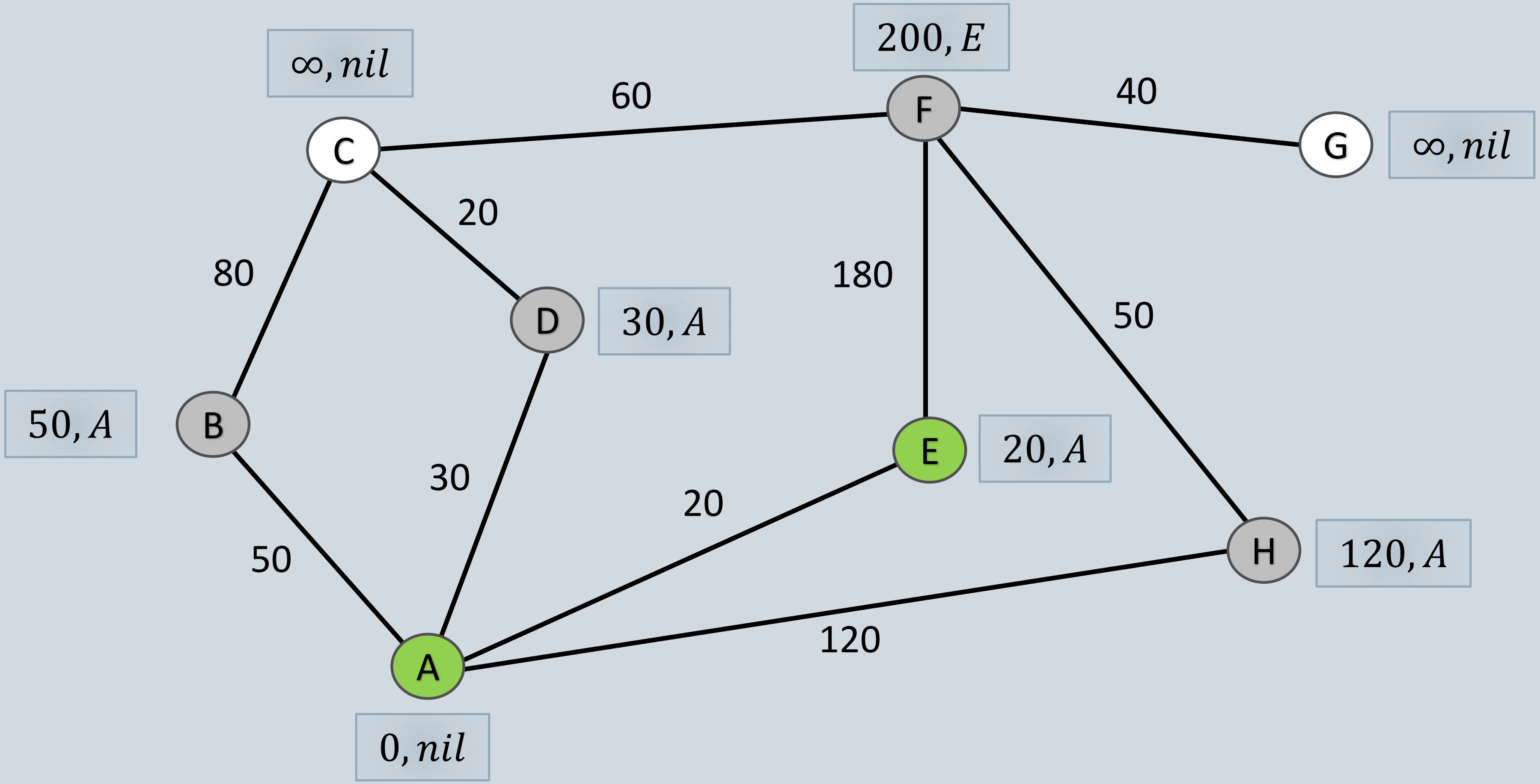
∞, nil
60
F
40
∞, nil
C
G
∞, nil
20
80
180
D
∞, nil
50
∞, nil
B
E
∞, nil
30
20
50
H
∞, nil
120
A
0, nil

Exemple



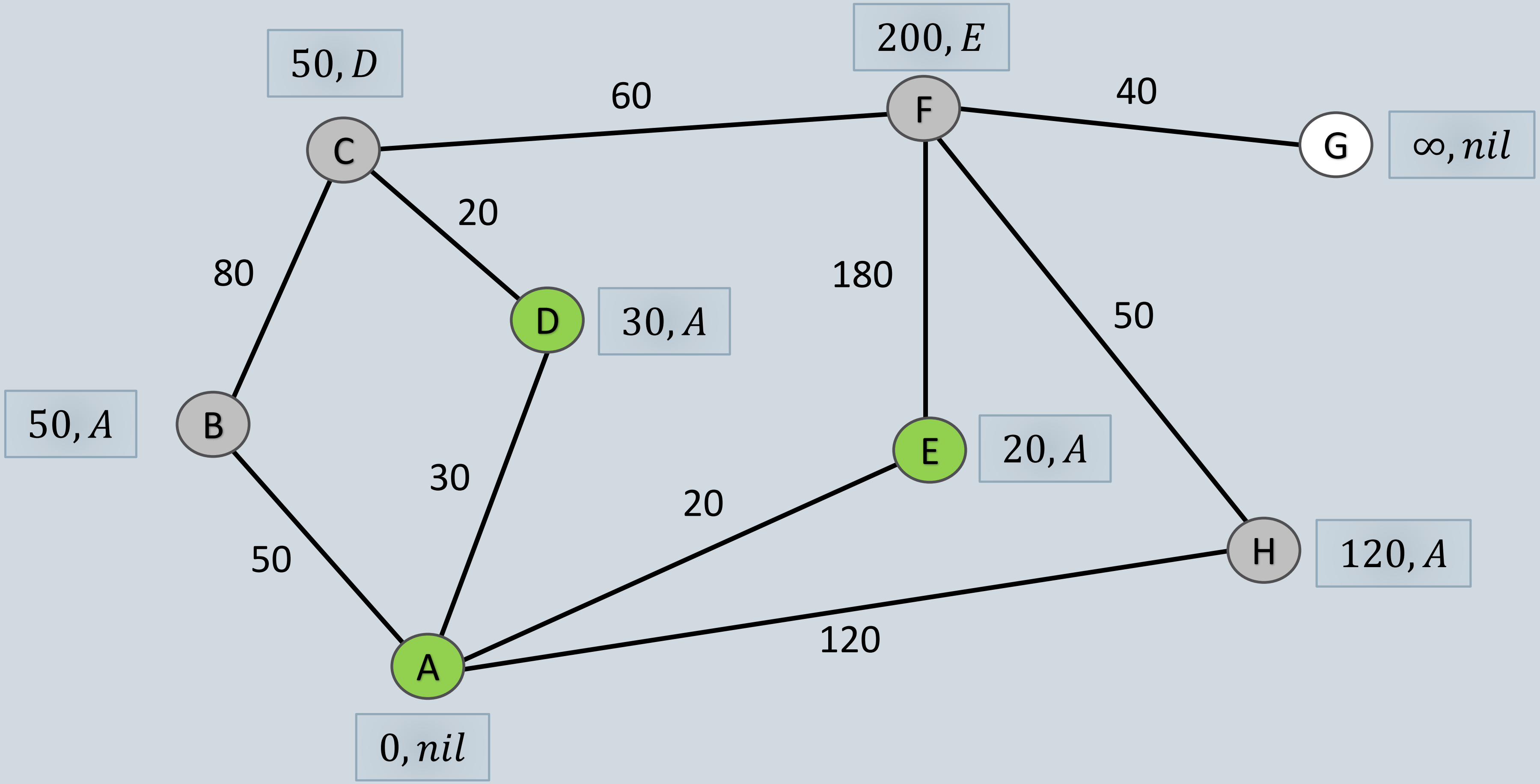
∞, nil
∞, nil
60
40
F
C
G
∞, nil
20
80
180
D
$30, A$
50
$50, A$
B
E
$20, A$
30
20
50
H
$120, A$
120
A
$0, nil$

Exemple



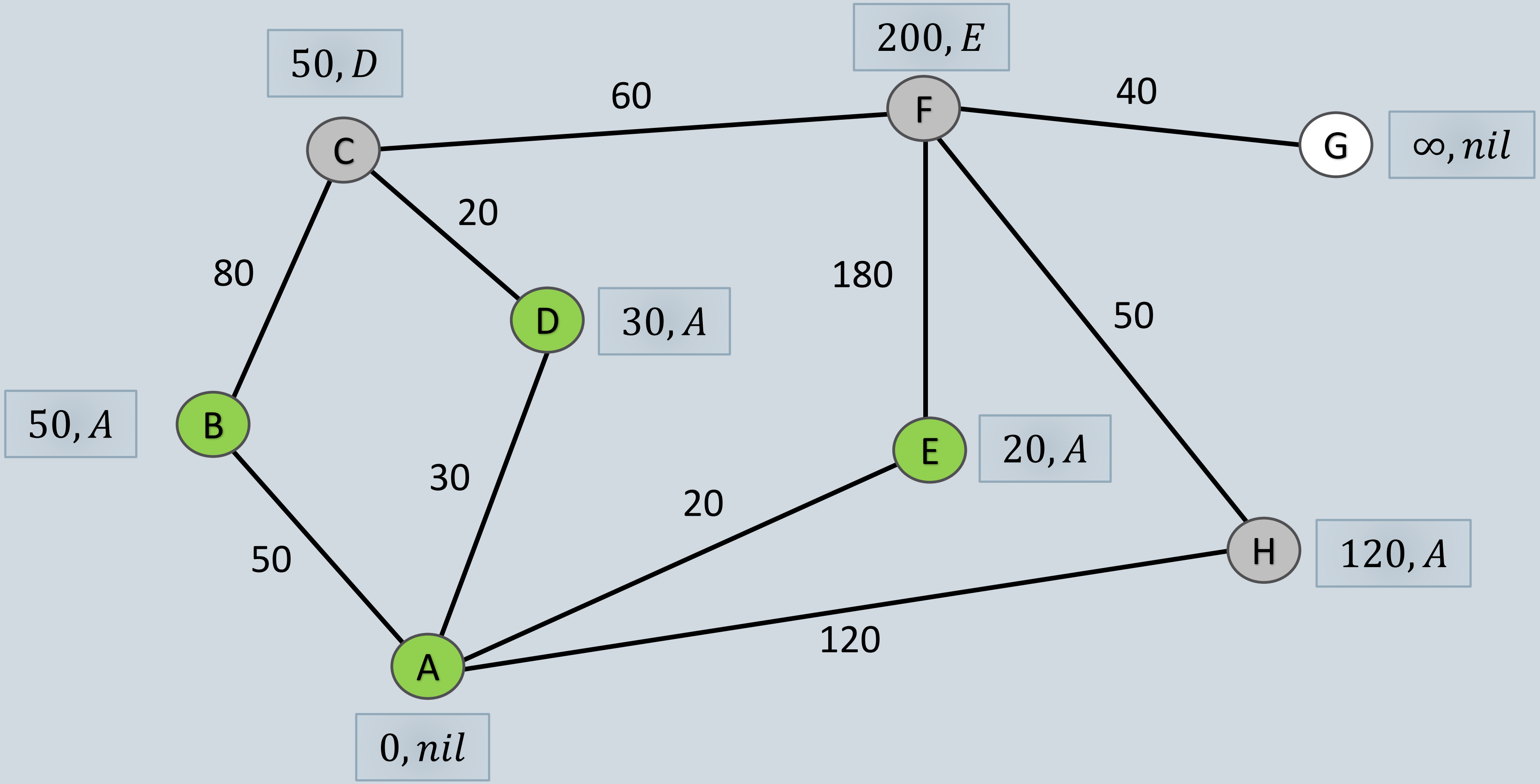
∞, nil
60
$200, E$
40
C
F
G
∞, nil
20
80
180
50
D
$30, A$
$50, A$
B
E
$20, A$
30
20
50
H
$120, A$
A
120
$0, nil$

Exemple



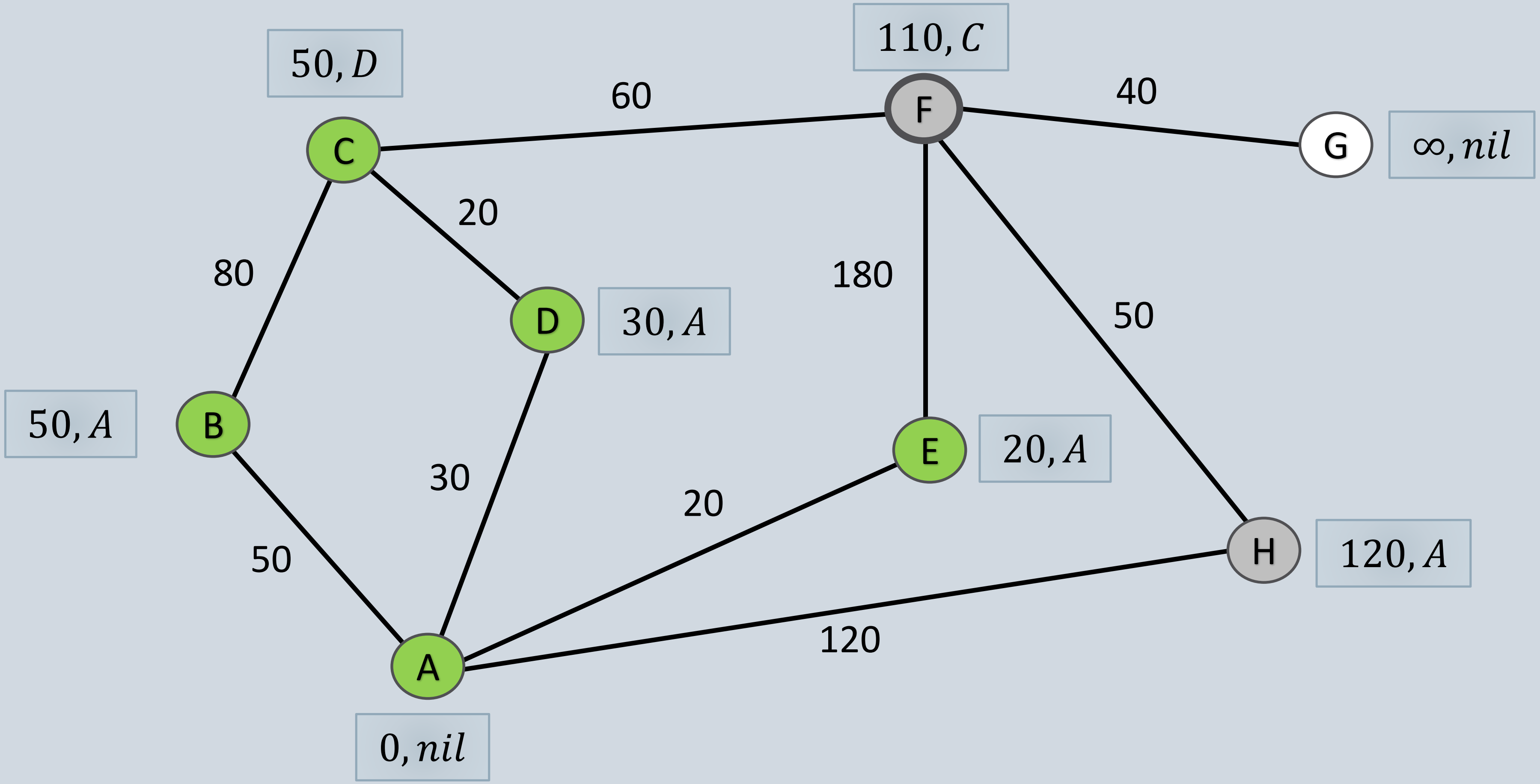
50, D
60
200, E
F
40
G
∞, nil
C
20
80
D
30, A
180
50
50, A
B
E
20, A
30
20
50
H
120, A
A
120
0, nil

Exemple



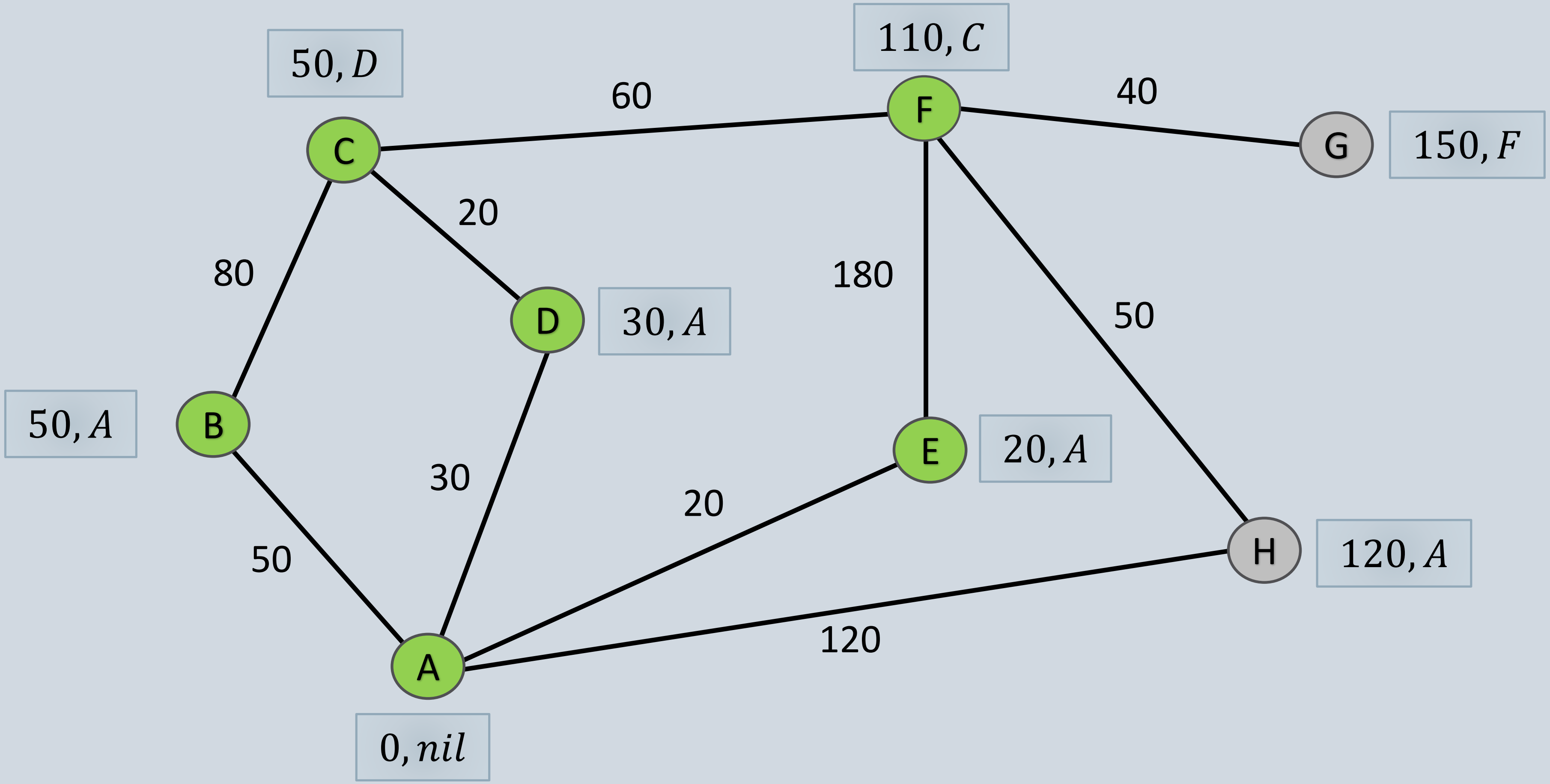
$50, D$
60
$200, E$
40
F
G
∞, nil
C
20
80
180
50
D
$30, A$
$50, A$
B
E
$20, A$
30
20
50
H
$120, A$
A
120
$0, nil$

Exemple



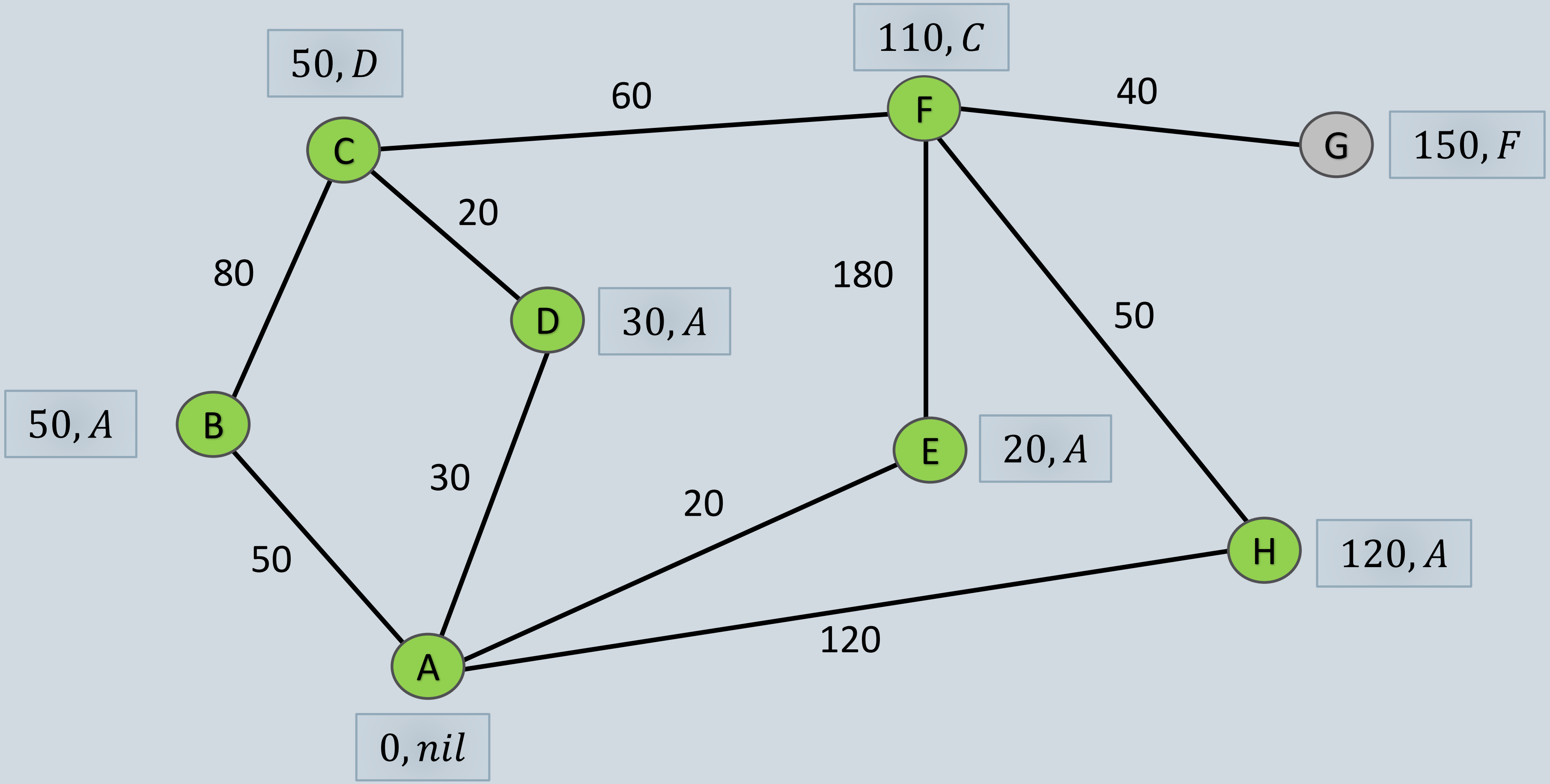
$110, C$
$50, D$
60
F
40
G
∞, nil
C
20
80
180
D
$30, A$
50
$50, A$
B
E
$20, A$
30
20
50
H
$120, A$
120
A
$0, nil$

Exemple



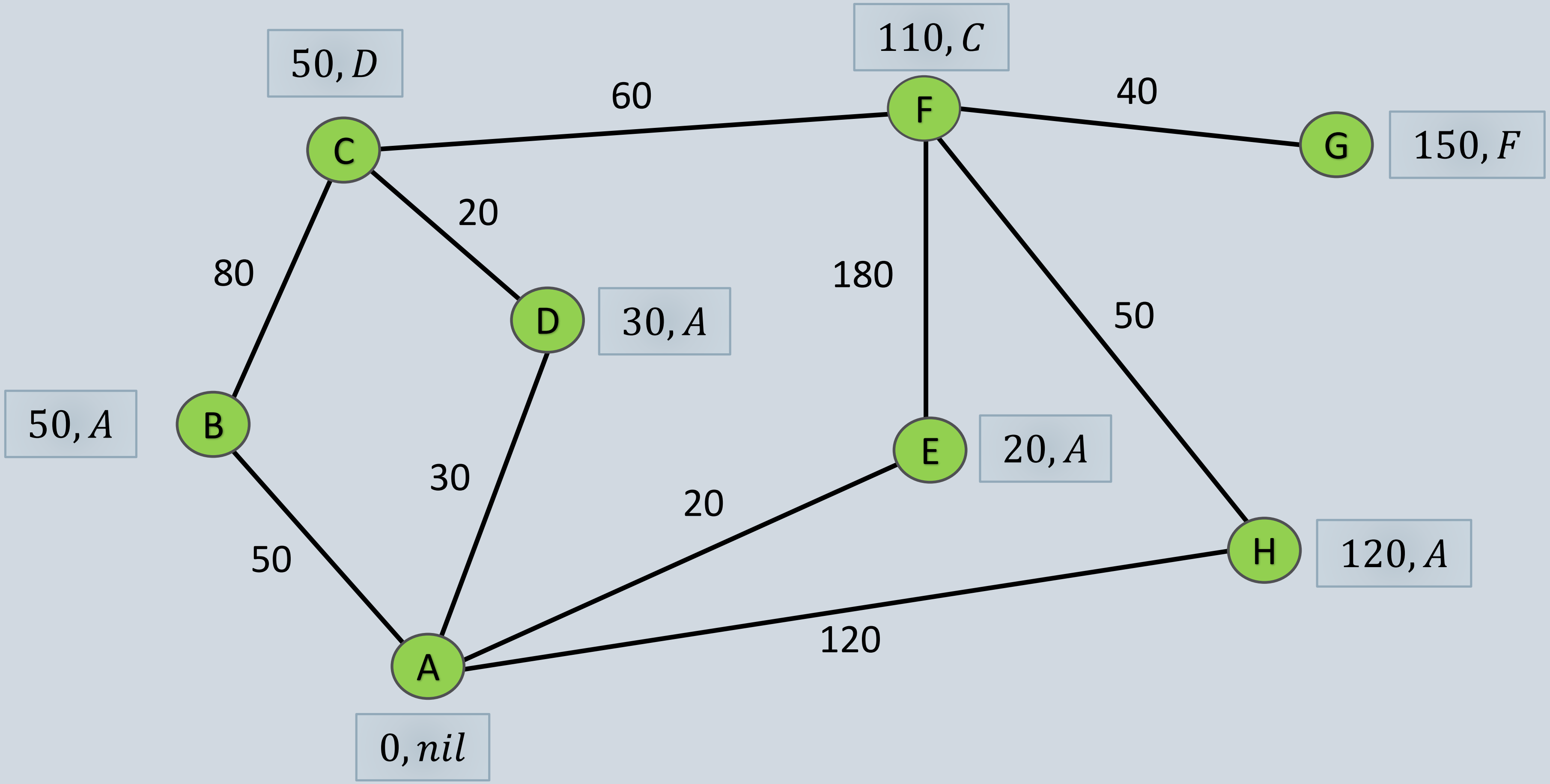
$50, D$
60
$110, C$
40
F
C
G
$150, F$
20
80
180
D
$30, A$
50
$50, A$
B
E
$20, A$
30
20
H
$120, A$
50
A
120
$0, nil$

Exemple



$50, D$
60
$110, C$
40
C
F
G
$150, F$
20
80
180
D
$30, A$
50
$50, A$
B
E
$20, A$
30
20
H
$120, A$
50
A
120
$0, nil$

Exemple



$50, D$
60
$110, C$
40
F
C
G
$150, F$
20
80
180
D
$30, A$
50
$50, A$
B
E
$20, A$
30
20
H
$120, A$
50
120
A
$0, nil$

Complexité

Implémentation du tas (Priority Queue)	poll	update	|V| x poll + (|V|+|E|) x update
Tableau / Liste	$O(\|V\|)$	$O(1)$	$O(\|V\|^2)$
Tas binaire	$O(\log \|V\|)$	$O(\log \|V\|)$	$O\big((\|V\| + \|E\|)\log \|V\|\big)$
Tas n-aire	$O\left(\frac{n \log \|V\|}{\log n}\right)$	$O\left(\frac{\log \|V\|}{\log n}\right)$	$O\left((\|V\|.n + \|E\|)\frac{\log \|V\|}{\log n}\right)$
Tas de Fibonacci	$O(\log \|V\|)$	$O(1)$ amorti	$O(\|V\| \log \|V\| + \|E\|)$

Importance des concepts acquis lors du module M313 sur les structures de données!
Implémentation des collections ordonnées : voir par exemple l'animation du cours https://www.cs.usfca.edu/~galles/visualization/RedBlack.html

Implémentation Java

http://docs.oracle.com/javase/tutorial/collections/interfaces/queue.html

```java
public class Dijkstra {
    public static void computePaths(Vertex source) {
        source.minDistance = 0.;
        PriorityQueue<Vertex> vertexQueue = new PriorityQueue<Vertex>();
        vertexQueue.add(source);

        while (!vertexQueue.isEmpty()) {
            Vertex u = vertexQueue.poll();

            for (Edge e : u.adjacencies) {// Visit each edge exiting u
                Vertex v = e.target;
                double weight = e.weight;
                double distanceThroughU = u.minDistance + weight;
                if (distanceThroughU < v.minDistance) {
                    vertexQueue.remove(v);
                    v.minDistance = distanceThroughU;
                    v.previous = u;
                    vertexQueue.add(v);
                }
            }
        }
    }
```