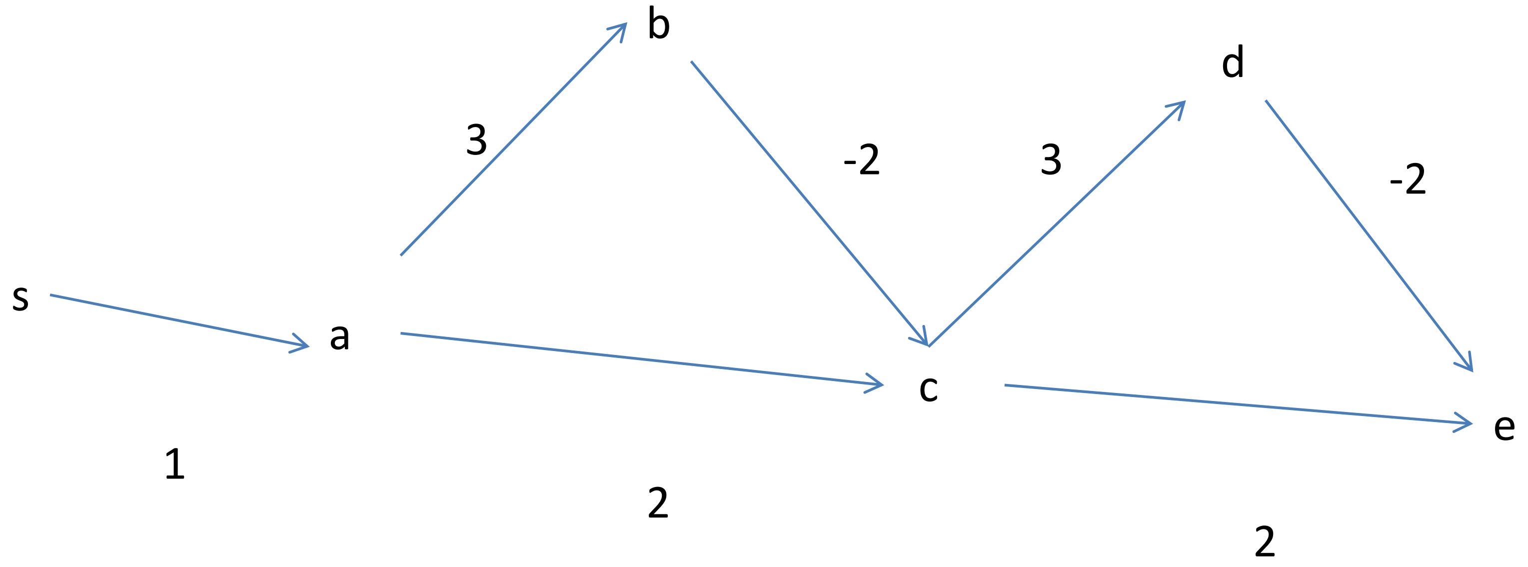
ALGORITHMES DE RECHERCHE DES CHEMINS DE POIDS MINIMAUX À ORIGINE UNIQUE

SECONDE PARTIE

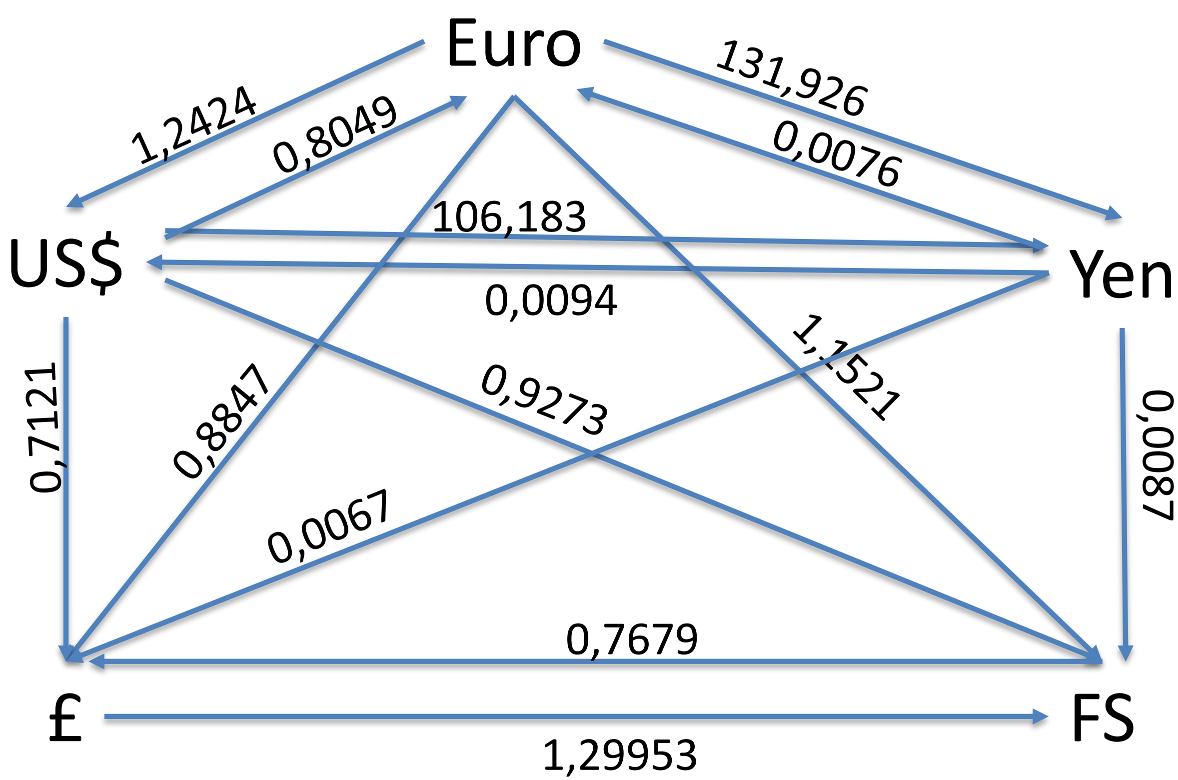
Arcs de pondération négative ?



Pondérations négatives ?
Un petit exemple

- Problème d'arbitrage de change :
  - Données :
    - GTC : un graphe de taux de change
    - S : une monnaie de départ
    - D : Une monnaie d'arrivée
  - Question : quel chemin de S à D me permet d'optimiser la somme obtenue dans la devise D ?

Pondérations négatives ?
Un petit exemple



Pondérations négatives ?
Un petit exemple

- Pour 100 Euros j'aurai 115,21 francs suisses
- Pour 100 Euros j'aurai 13192,6 Yen
- Pour 13192,6 Yen j'aurai 13192,6 * 0,0094 $
- Pour 13192,6 * 0,0094 $ j'aurai

  13192,6 * 0,0094 * 0,9273 Francs suisses
- Avons-nous :

  115,21 = 13192,6 * 0,0094 * 0,9273 ?

Pondérations négatives ? Un petit exemple

- On cherche un chemin avec un poids mais ce poids est calculé par le produit des poids des arcs (au lieu de la somme).
- On cherche un chemin de poids maximal (au lieu de minimal.

Pondérations négatives ?
Un petit exemple

- Remarque 1 : $0 < x < y \Rightarrow 0 < 1/y < 1/x$
- Remarque 2 : $c = a*b \Rightarrow 1/c = 1/a * 1/b$
- Remarque 3 : $\log(a*b*c) = \log(a)+\log(b)+\log(c)$
- Remarque 4 : log est une fonction croissante sur l'intervalle $]0, +\infty[$.

Pondérations négatives ? Un petit exemple

- On souhaite un chemin de poids maximal.
- Pour un chemin $\mu=x_1x_2x_3\ldots x_k$ de notre graphe des taux de change le poids de ce chemin est : $V(\mu)=V(x_1x_2)*V(x_2x_3)*\ldots*V(x_{k-1}x_k)$
- Remarque si touts les poids sont strictement positif $V(x_1x_2)*V(x_2x_3)*\ldots*V(x_{k-1}x_k)$ est maximal si et ssi $(1/V(x_1x_2))*(1/V(x_2x_3))*\ldots*(1/V(x_{k-1}x_k))$ est minimal.

Pondérations négatives ?
Un petit exemple

Nous créons une nouvelle pondération de nos arcs :

- $V'(xy) = 1/V(xy)$

Cette nouvelle pondération nous permet de transformer la recherche d'un chemin de poids maximal (en utilisant V) en la recherche d'un chemin de poids minimal (en utilisant V')

Pondérations négatives ?
Un petit exemple

Reste le problème de la multiplication que nous allons résoudre grâce à la remarque 3 :
log(a*b*c) = log(a)+log(b)+log(c)

Posons V’’(xy) = log (V’(xy))

Cette nouvelle pondération nous permet de transformer la recherche d’un chemin de poids minimal (en utilisant *) en la recherche d’un chemin de poids minimal (en utilisant +)

Pondérations négatives ?
Un petit exemple

- On transforme donc toutes les pondérations :
  - $V''(uv) = \log(1/V(uv)) = -\log(V(uv))$

- On cherche alors un chemin de poids minimal (au sens que nous avons donné dans la première partie) qui contiendra des arcs de pondérations négatives. Car $-\log(V(uv)) < 0$ dès que $V(uv) > 1$

Algorithme de Bellman-Ford

- Cet algorithme construit les chemins de poids minimaux en acceptant les arcs de pondération négative.

- Il faut donc accepter de reconsidérer des sommets pour lesquels un chemin à été calculé.

Principes de Bellman-Ford

- Construire les chemins à partir du sommet s par ordre de longueur croissante.
- Retenir pour chaque sommet un chemin qui réalise la valeur minimale
- Initialement on sait construire un chemin de longueur 0 allant de s à s de poids nul.

Principes de Bellman-Ford

- Considérons que nous avons un tableau D (indicé par les sommets) de valeurs

- A l'étape i : $D_i[x]$ contient le poids minimal d'un chemin de longueur au plus i ayant pour origine s et pour extrémité x. Si un tel chemin n'existe pas $D[x] = \infty$
- Construisons D_{i+1} (D à l'étape i+1)

Principes de Bellman-Ford

- Un chemin de poids minimal de longueur au plus i+1 se décompose en un chemin de poids minimal de longueur au plus i (dans D_i) et un arc qui prolonge ce chemin.
- $D_{i+1}[x] = \mathrm{Min}(D_i[x], \mathrm{Min}_{y \in Pred(x)}(D_i[y]+V(yx)))$

Condition finale

- Avoir construit tous les chemins élémentaires de poids potentiellement minimal. Or S’il n’y a pas de circuits de poids négatifs un chemin élémentaire est au plus de longueur $|X|$ - 1.

Algo BF (entête)

- Algo BF
- Données :
  - G = (X,U,V) un graphe pondéré
  - s un sommet de G
- Résultats :
  - GCPM = (X, U’, V) un graphe pondéré
  - D : tableau de nombres indicé par les sommets
- Variables :
  - Origine : tableau de sommets indicé par les sommets

Algo BF (code 1)

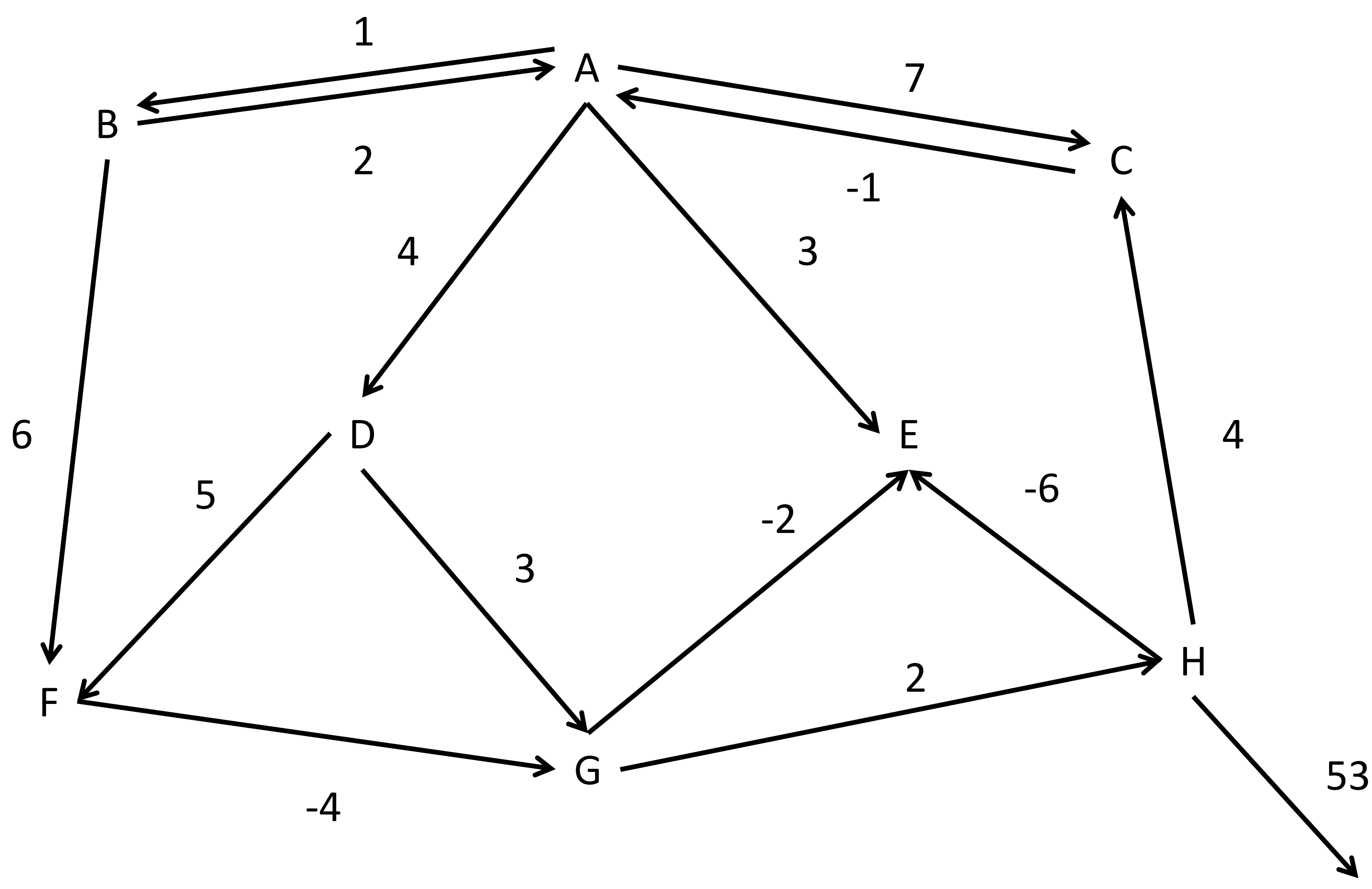
- Début
  - Pour chaque sommet t de X faire
    - D[t] ← ∞; Origine [t] ← t
  - FinPour
  - D[s] ← 0; U’ ← {};

Algo BF (code 2)

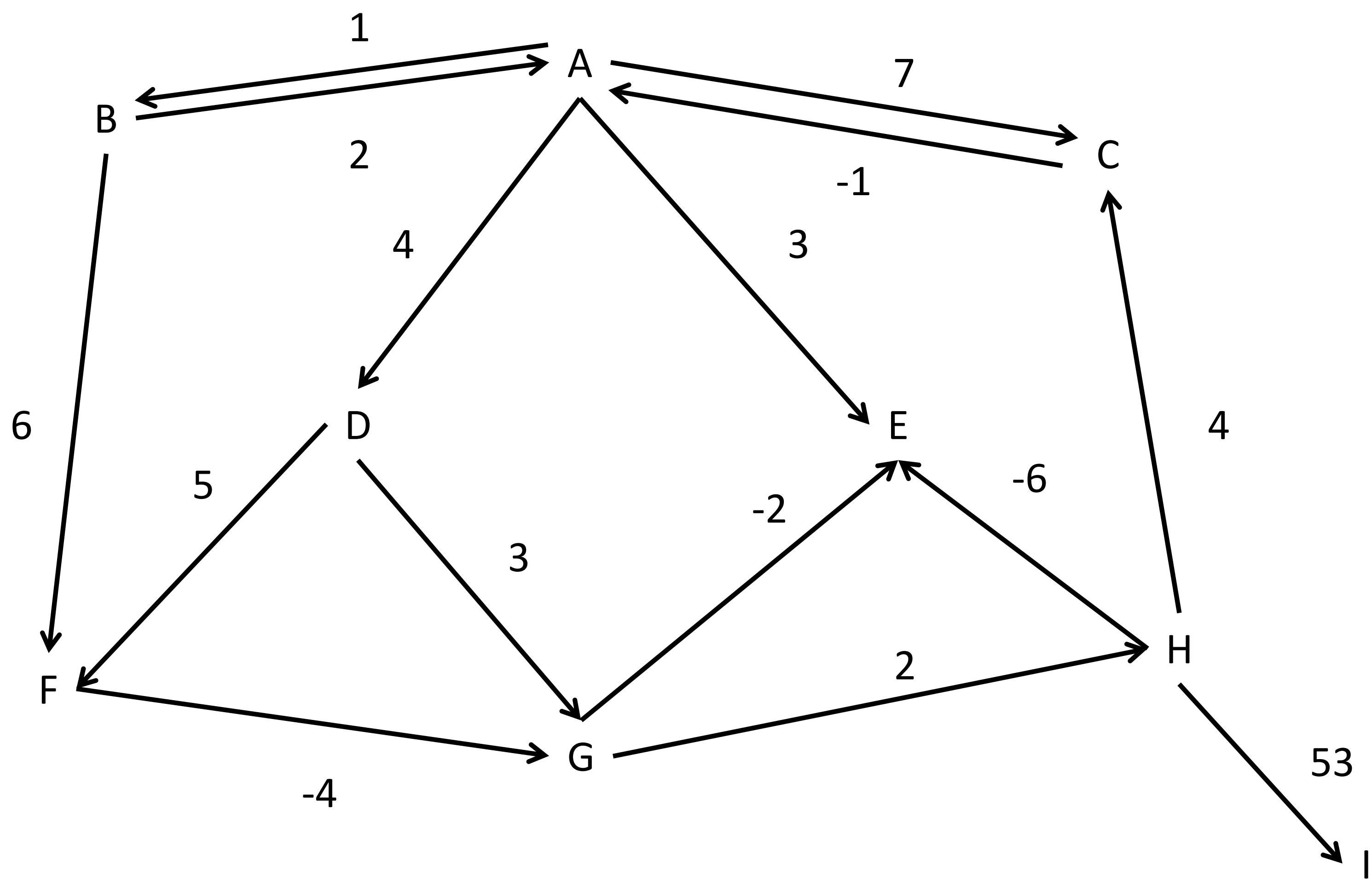
- Pour i allant de 1 à |X|- 1 faire
  - Pour chaque sommet u ϵ X faire
    - Si D[u] ≠ ∞ alors
      - » Pour chaque y ϵ Succ(u) faire
        - Si D[y] > D[u] + v(uy) alors
          - D[y] ← D[u] + v(uy); Origine [y] ← u;
        - FinSi
      - » FinPour
    - FinSi
  - FinPour
- FinPour
- Pour chaque sommet u ϵ X-{s} faire
  - Insérer l’arc (Origine[u], u) dans U’
- FinPour

- FinCode

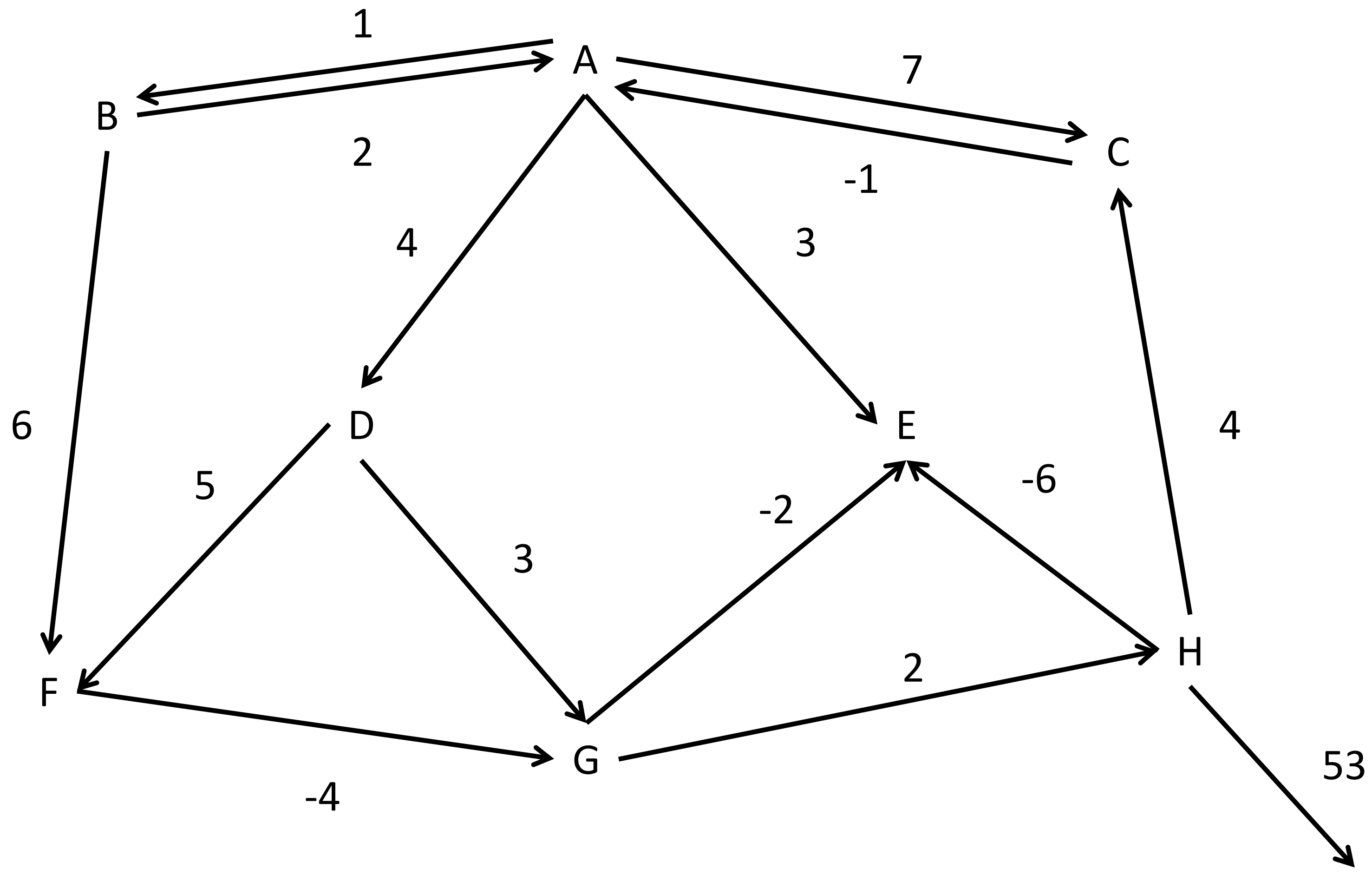
Exemple



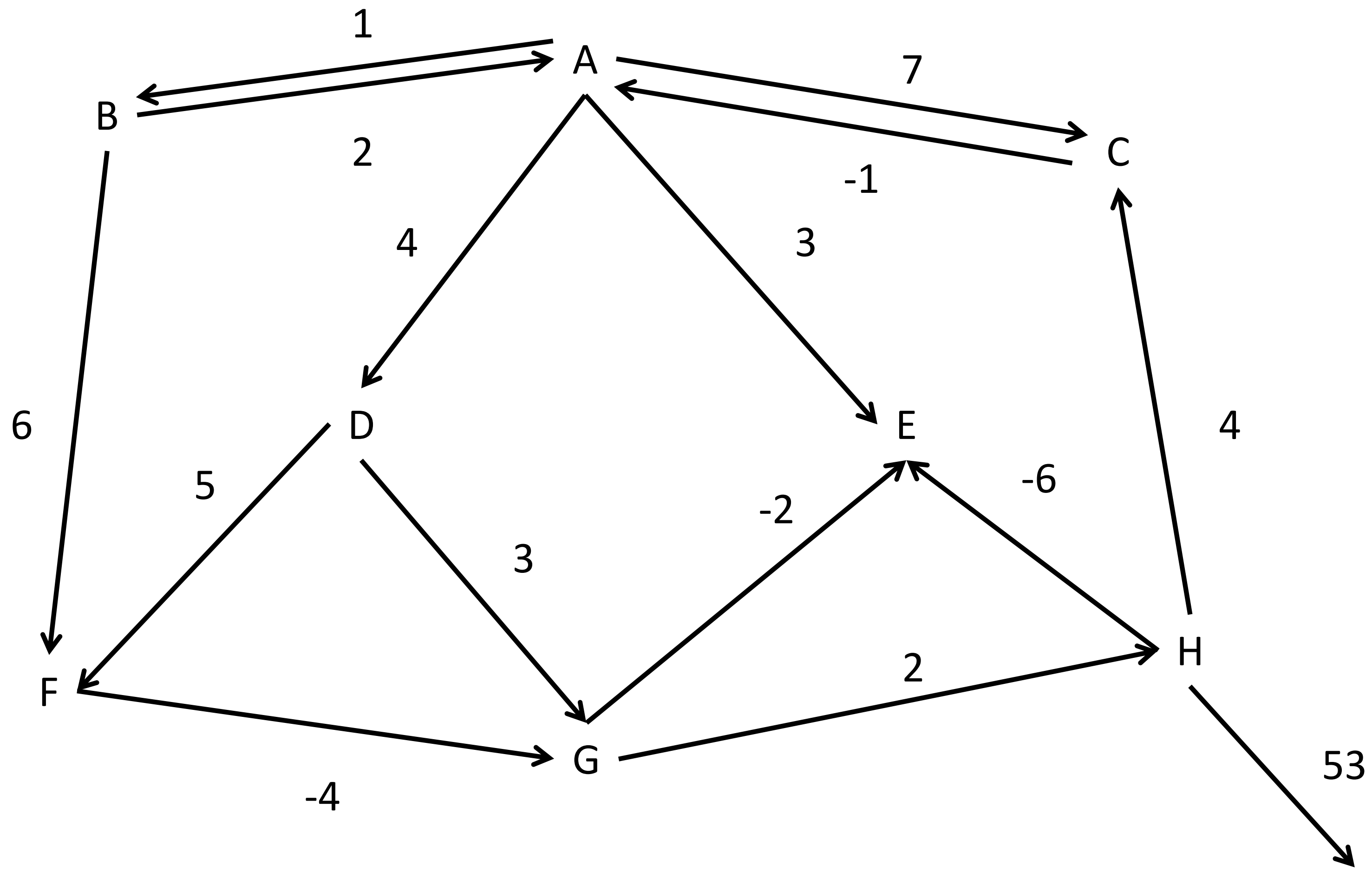
A	B	C	D	E	F	G	H	I
0	∞	∞	∞	∞	∞	∞	∞	∞
A	B	C	D	E	F	G	H	I



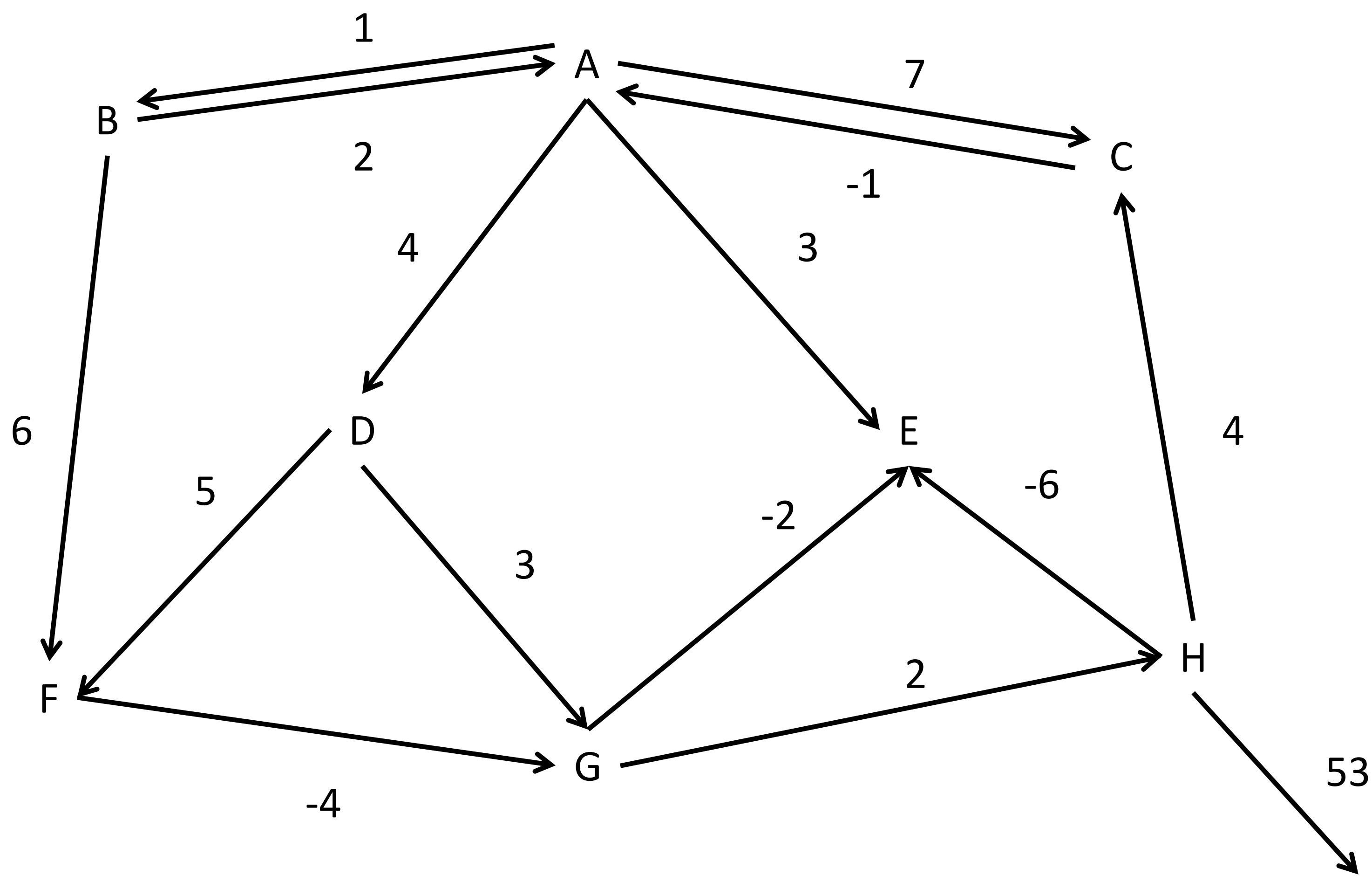
A	B	C	D	E	F	G	H	I
0	1	7	4	3	∞	∞	∞	∞
A	A	A	A	A	F	G	H	I



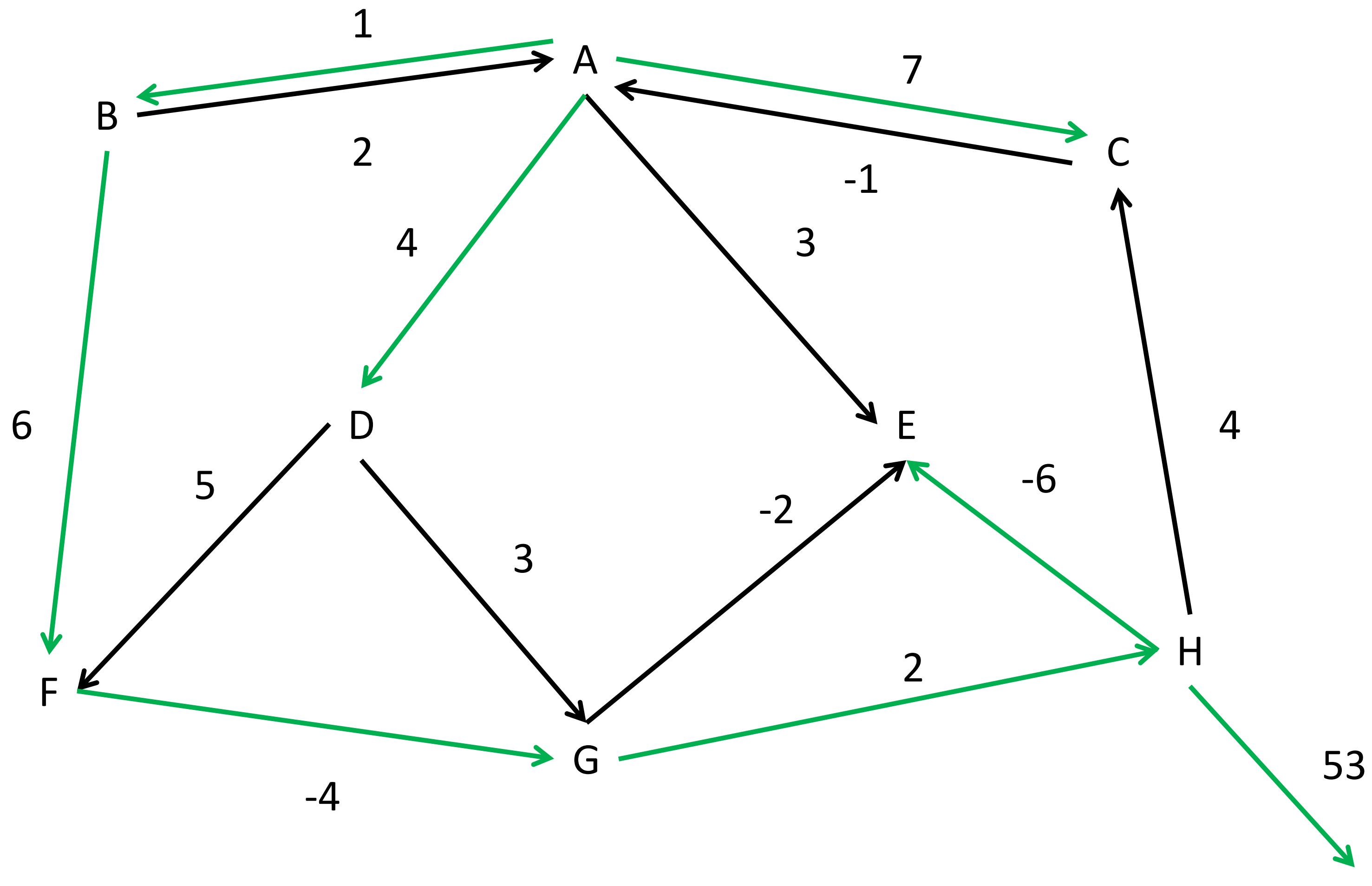
A	B	C	D	E	F	G	H	I
0	1	7	4	3	7	7	∞	∞
A	A	A	A	A	B	D	H	I



A	B	C	D	E	F	G	H	I
0	1	7	4	-1	7	3	5	58
A	A	A	A	H	B	F	G	H



A	B	C	D	E	F	G	H	I
0	1	7	4	-1	7	3	5	58
A	A	A	A	H	B	F	G	H



Propriété

- Soit $\mu = x_1, x_2, x_3, \ldots, x_{k-1}, x_k$ un chemin de poids minimal de x_1 à x_k du graphe pondéré G = (X, U, V) alors : $\mu' = x_1, x_2, x_3, \ldots, x_{k-1}$ est un chemin de poids minimal de x_1 à x_{k-1}.

- Bellman-Ford exploite cette propriété en effet à l'étape k cet algorithme construit les chemins de poids minimaux de longueur inférieure ou égale à k.

Complexité

- La complexité de cet algorithme est :
  - $O(n^3)$ pour une représentation par matrice
  - $O(nm)$ pour une représentation par listes d'adjacence