# PROBLEME DU VOYAGEUR DE COMMERCE
# ALGORITHME DE LITTLE

On s'intéresse à un voyageur de commerce qui doit visiter chacune des 6 villes suivantes (ABCDEF) une fois et une seule et revenir à son point de départ.

Un tableau des distances intervilles étant donné, quelle sera la tournée la moins longue ?

**Algorithme de LITTLE**

Réduction de la matrice (/ligne et /colonne cf Méthode hongroise)
Choisir la case nulle dont le coût d'éviction est le plus élevé.
Le coût d'éviction est le surcoût minimal engendré par le fait de ne pas choisir la case, c'est à dire choisir une autre case sur la même ligne et une sur la même colonne…
Enlever la ligne de la ville de départ et la colonne de la ville d'arrivée correspondantes.
Rendre infini le coût de retour
Recommencer avec le tableau partiel….
Présenter l'algorithme sous la forme d'une arborescence (Branch and Bound, Séparation-Evaluation).
Examiner tous les chemins dont le coût est inférieur au premier coût final trouvé.
Dérouler l'arborescence jusqu'au bout.

Exemple :

| ↗ | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| A | ∞ | 1 | 7 | 3 | 14 | 2 |
| B | 3 | ∞ | 6 | 9 | 1 | 24 |
| C | 6 | 14 | ∞ | 3 | 7 | 3 |
| D | 2 | 3 | 5 | ∞ | 9 | 11 |
| E | 15 | 7 | 11 | 2 | ∞ | 4 |
| F | 20 | 5 | 13 | 4 | 18 | ∞ |

| ↗ | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| A | ∞ | 0 | 6 | 2 | 13 | 1 |
| B | 2 | ∞ | 5 | 8 | 0 | 23 |
| C | 3 | 11 | ∞ | 0 | 4 | 0 |
| D | 0 | 1 | 3 | ∞ | 7 | 9 |
| E | 13 | 5 | 9 | 0 | ∞ | 2 |
| F | 16 | 1 | 9 | 0 | 14 | ∞ |

| ↗ | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| A | ∞ | 0 | 3 | 2 | 13 | 1 |
| B | 2 | ∞ | 2 | 8 | 0 | 23 |
| C | 3 | 11 | ∞ | 0 | 4 | 0 |
| D | 0 | 1 | 0 | ∞ | 7 | 9 |
| E | 13 | 5 | 6 | 0 | ∞ | 2 |
| F | 16 | 1 | 6 | 0 | 14 | ∞ |

Il faudra rajouter 16 au coût

**Les coûts d'éviction sont indiqués entre ( )**

| ↗ | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| A | ∞ | 0,(2) | 3 | 2 | 13 | 1 |
| B | 2 | ∞ | 2 | 8 | 0,(6) | 23 |
| C | 3 | 11 | ∞ | 0,(0) | 4 | 0,(1) |
| D | 0,(2) | 1 | 0,(2) | ∞ | 7 | 9 |
| E | 13 | 5 | 6 | 0,(2) | ∞ | 2 |
| F | 16 | 1 | 6 | 0,(1) | 14 | ∞ |

On retiendra (B,E) dont le coût d'éviction est de 6, le coût minimal des tournées sans (B,E) est donc de 22.
On supprime la ligne B et la colonne E et on rend (E,B) infini :

| ↗ | A | B | C | D | F |
|---|---|---|---|---|---|
| A | ∞ | 0,(2) | 3 | 2 | 1 |
| C | 3 | 11 | ∞ | 0,(0) | 0,(1) |
| D | 0,(**3**) | 1 | 0,(**3**) | ∞ | 9 |
| E | 13 | ∞ | 6 | 0,(2) | 2 |
| F | 16 | 1 | 6 | 0,(1) | ∞ |

Ce tableau est réduit (un zéro par ligne et par colonne), on sait donc que les tournées avec (B,E) auront un coût minimal ne dépassant pas 16.

Sans (B,E) = 22 | Avec (B,E) = 16

On recommence sur le tableau partiel : entre (D,A) et (D,C) on choisit **arbitrairement** le premier (D,A).

| ↗ | B | C | D | F |
|---|---|---|---|---|
| A | 0,(**2**) | 3 | ∞ | 1 |
| C | 11 | ∞ | 0,(0) | 0,(1) |
| E | ∞ | 6 | 0,(2) | 2 |
| F | 1 | 6 | 0,(1) | ∞ |

Ce tableau n'est pas réduit, pour le réduire, il faut enlever 3 dans la colonne C.

Sans (B,E) = 22 | Avec (B,E) = 16

Sans (D,A) = 19 | Avec (D,A) = 19

Il faudra explorer la branche avec (D,A) et la branche sans (D,A). Commençons par celle avec (D,A) :

| ↗ | B | C | D | F |
|---|---|---|---|---|
| A | 0,(1) | 0,(**3**) | ∞ | 1 |
| C | 11 | ∞ | 0,(0) | 0,(1) |
| E | ∞ | 3 | 0,(2) | 2 |
| F | 1 | 3 | 0,(1) | ∞ |

On choisira donc (A,C). Il ne faudra pas oublier de rendre le coût de C à D infini.

| ↗ | B | D | F |
|---|---|---|---|
| C | 11 | ∞ | 0 |
| E | ∞ | 0 | 2 |
| F | 1 | 0 | ∞ |

| ↗ | B | D | F |
|---|---|---|---|
| C | 10 | ∞ | 0,(12) |
| E | ∞ | 0,(2) | 2 |
| F | 0,(10) | 0,(1) | ∞ |

Sans (B,E) = 22 | Avec (B,E) = 16

Sans (D,A) = 19 | Avec (D,A) = 19

Sans (A,C) = 22 | Avec (A,C) = 20

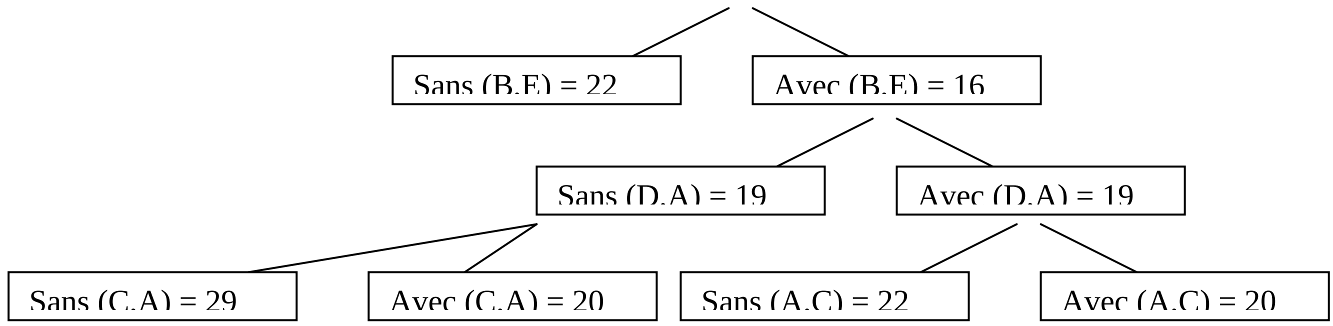
Le coût du chemin avec BE, DA, AC étant de 20 supérieur à 19 coût du chemin BE, sans DA, il nous faut examiner cette branche : pour cela, il faut reprendre le tableau avant élimination de la ligne D et la colonne A et rendre infini le coût de D à A, soit :

| ↗ | A | B | C | D | F |
|---|---|---|---|---|---|
| A | ∞ | 0 | 3 | 2 | 1 |
| C | 3 | 11 | ∞ | 0 | 0 |
| D | ∞ | 1 | 0 | ∞ | 9 |
| E | 13 | ∞ | 6 | 0 | 2 |
| F | 16 | 1 | 6 | 0 | ∞ |

| ↗ | A | B | C | D | F |
|---|---|---|---|---|---|
| A | ∞ | 0,(2) | 3 | 2 | 1 |
| C | 0,(10) | 11 | ∞ | 0,(0) | 0,(1) |
| D | ∞ | 1 | 0,(4) | ∞ | 9 |
| E | 10 | ∞ | 6 | 0,(2) | 2 |
| F | 13 | 1 | 6 | 0,(1) | ∞ |

| ↗ | B | C | D | F |
|---|---|---|---|---|
| A | 0,(2) | ∞ | 2 | 1 |
| D | 1 | 0,(4) | ∞ | 9 |
| E | ∞ | 6 | 0,(2) | 2 |
| F | 1 | 6 | 0,(1) | ∞ |

Ce tableau doit être réduit pour poursuivre, donc les solutions avec CA auront un coût minimum de 20.

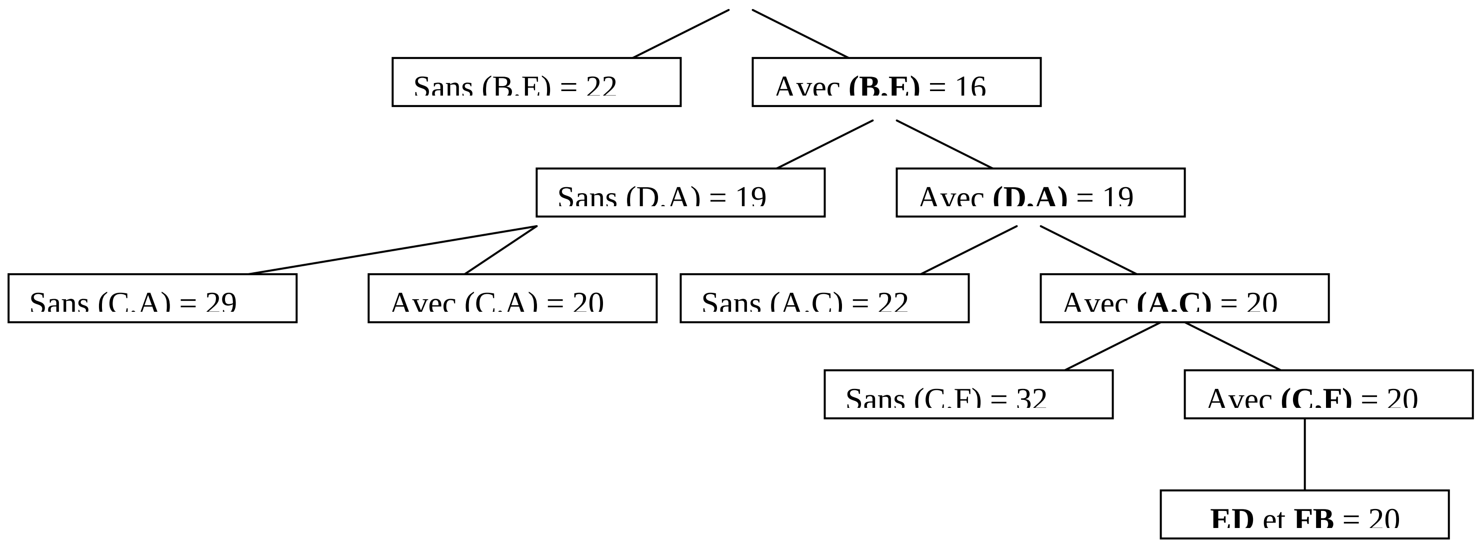



Comme aucune branche à gauche ne vaut moins de 20, on peut choisir de reprendre l'exploration à droite.

On choisira CF.

| ↗ | B | D |
|---|---|---|
| E | ∞ | 0 |
| F | 0 | ∞ |

On pourra rajouter ED et FB sans accroître le coût, d'où l'arborescence finale :




La solution finale est donc : BEDACFB d'un coût total de 20.