

Recherche Opérationnelle
Épisode III
Arbres couvrants et ordonnancement

Hervé Hocquard

Université de Bordeaux, France

February 18, 2023

1. Arbres couvrants de poids minimum

Arbres

Arbre = Graphe **connexe** sans cycle.

Sous-graphe = Mêmes sommets, mais un sous-ensemble des arêtes.

Poids d'un sous-graphe = Somme des poids sur ses arêtes.

Arbre couvrant = Sous-graphe qui est un arbre.

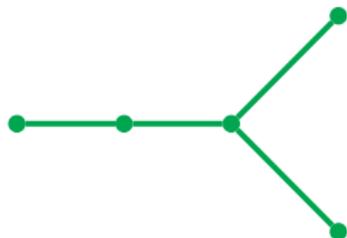
Arbres

Arbre = Graphe **connexe** sans cycle.

Sous-graphe = Mêmes sommets, mais un sous-ensemble des arêtes.

Poids d'un sous-graphe = Somme des poids sur ses arêtes.

Arbre couvrant = Sous-graphe qui est un arbre.



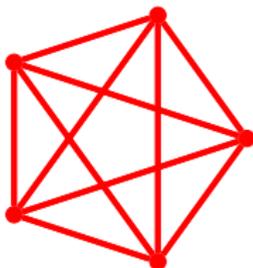
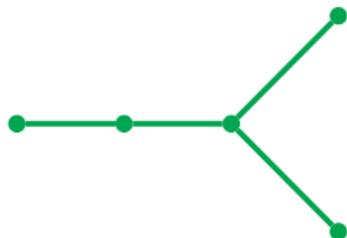
Arbres

Arbre = Graphe **connexe** sans cycle.

Sous-graphe = Mêmes sommets, mais un sous-ensemble des arêtes.

Poids d'un sous-graphe = Somme des poids sur ses arêtes.

Arbre couvrant = Sous-graphe qui est un arbre.



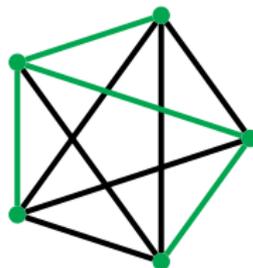
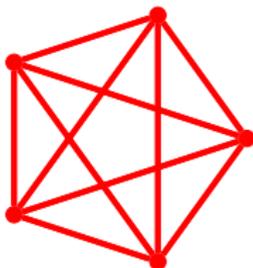
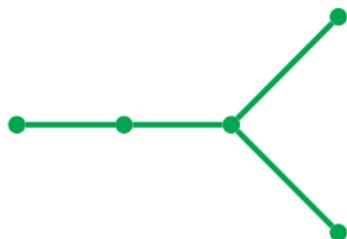
Arbres

Arbre = Graphe **connexe** sans cycle.

Sous-graphe = Mêmes sommets, mais un sous-ensemble des arêtes.

Poids d'un sous-graphe = Somme des poids sur ses arêtes.

Arbre couvrant = Sous-graphe qui est un arbre.



Propriétés

- ① Soit G un graphe à n sommets et m arêtes. On a :
- si G est connexe, alors $m \geq n - 1$;
 - si G est acyclique, alors $m \leq n - 1$.

Propriétés

- 1 Soit G un graphe à n sommets et m arêtes. On a :
 - si G est connexe, alors $m \geq n - 1$;
 - si G est acyclique, alors $m \leq n - 1$.
- 2 Donc un arbre à n sommets a exactement $n - 1$ arêtes...

Propriétés

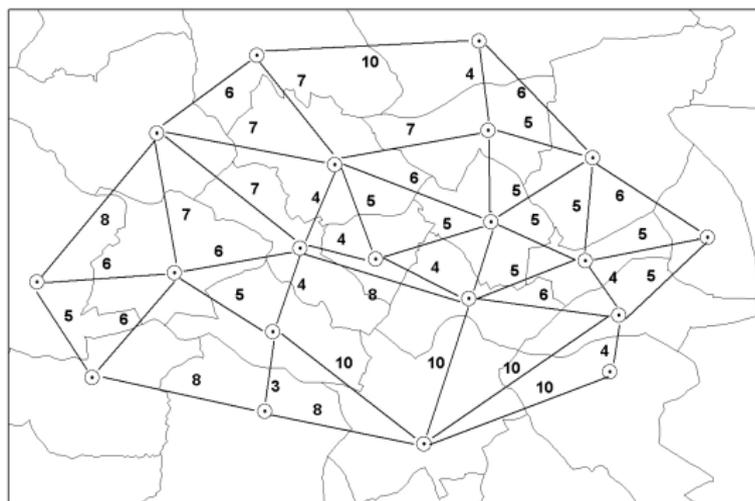
- 1 Soit G un graphe à n sommets et m arêtes. On a :
 - si G est connexe, alors $m \geq n - 1$;
 - si G est acyclique, alors $m \leq n - 1$.
- 2 Donc un arbre à n sommets a exactement $n - 1$ arêtes...
- 3 ... et tout ajout d'arête crée un cycle.

Propriétés

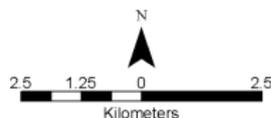
- 1 Soit G un graphe à n sommets et m arêtes. On a :
 - si G est connexe, alors $m \geq n - 1$;
 - si G est acyclique, alors $m \leq n - 1$.
- 2 Donc un arbre à n sommets a exactement $n - 1$ arêtes...
- 3 ... et tout ajout d'arête crée un cycle.
- 4 Tout graphe connexe admet un arbre couvrant.

Trouver un arbre couvrant de poids minimum

Intérêt : Construire un graphe connexe en minimisant un certain coût.

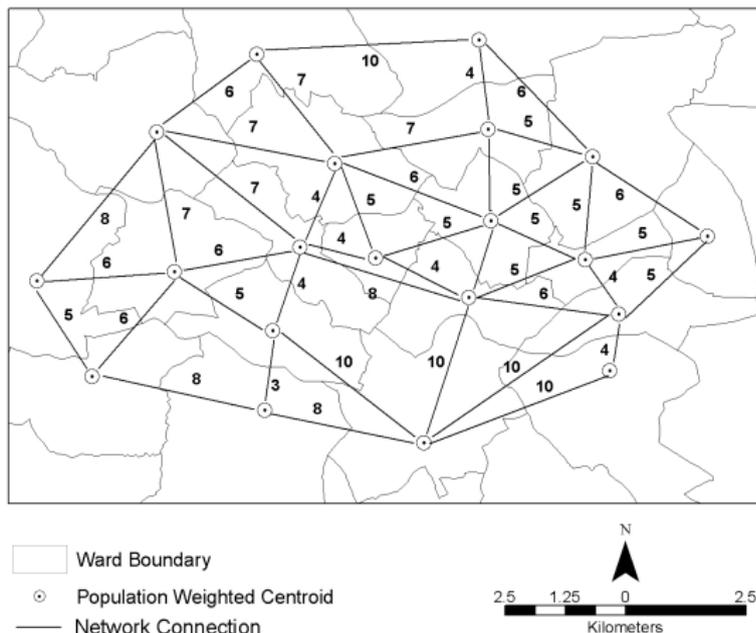


- Ward Boundary
- ⊙ Population Weighted Centroid
- Network Connection



Trouver un arbre couvrant de poids minimum

Intérêt : Construire un graphe connexe en minimisant un certain coût.



Deux algorithmes principaux : Prim et Kruskal.

Algorithme de Prim

Idée : Faire “grossir” l’arbre T depuis un sommet r , en ajoutant itérativement une arête de poids minimum adjacente à T , et ne créant pas de cycle.

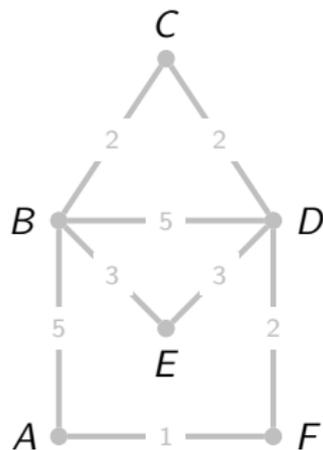
Algorithme de Prim

Idée : Faire “grossir” l’arbre T depuis un sommet r , en ajoutant itérativement une arête de poids minimum adjacente à T , et ne créant pas de cycle.

Algorithme de Prim :

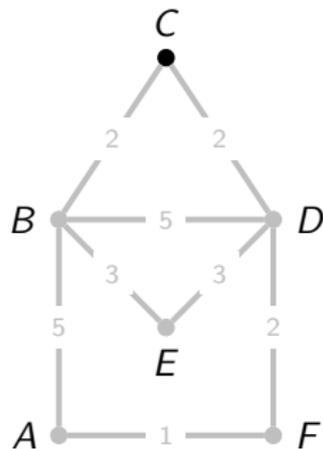
- **Entrée :** Un graphe pondéré G .
- **Initialisation :**
 - On choisit un sommet r de G .
 - L’arbre couvrant T est vide.
- **Traitement :**
 - Repérer toutes les arêtes e_1, \dots, e_k de G qui relie un sommet de T à un sommet extérieur à T .
 - Choisir parmi ces arêtes celle e_i qui a le plus petit poids.
 - Ajouter e_i à T (ainsi que le nouveau sommet atteint).
 - On s’arrête dès que tous les sommets de G sont dans T .
- **Sortie :** L’arbre couvrant T de G .

Exemple



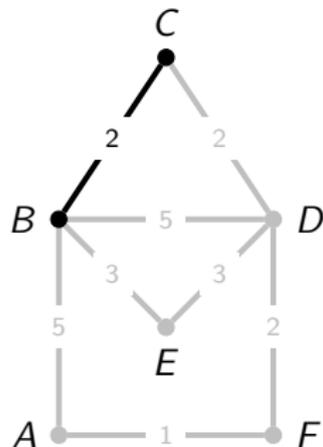
Poids de $T =$

Exemple



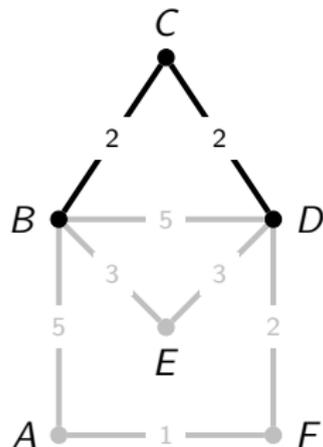
Poids de $T = 0$

Exemple



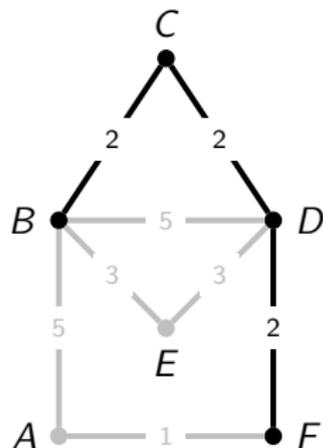
Poids de $T = 0 + 2$

Exemple



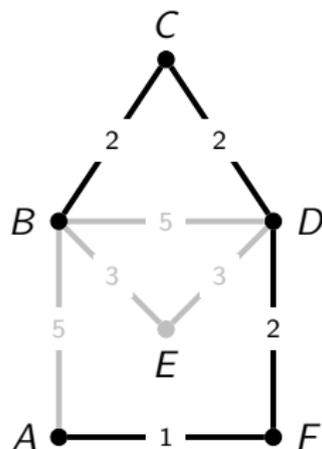
Poids de $T = 0 + 2 + 2$

Exemple



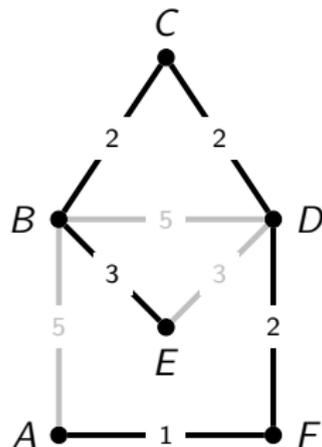
$$\text{Poids de } T = 0 + 2 + 2 + 2$$

Exemple



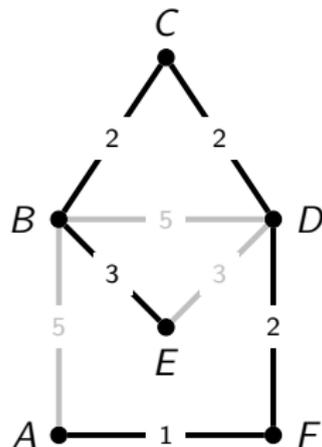
$$\text{Poids de } T = 0 + 2 + 2 + 2 + 1$$

Exemple



Poids de $T = 0 + 2 + 2 + 2 + 1 + 3$

Exemple



$$\text{Poids de } T = 0 + 2 + 2 + 2 + 1 + 3 = 10$$

Algorithme de Kruskal

Idée : On trie les arêtes par ordre croissant de poids, et on les ajoute à T , suivant leur ordre, tant qu'on ne crée pas de cycle.

Algorithme de Kruskal

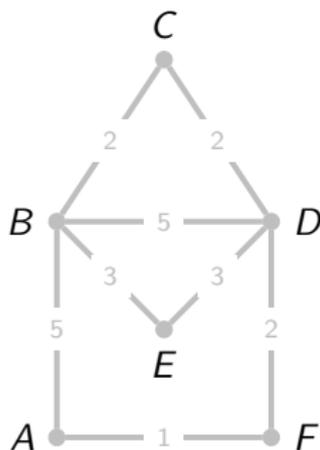
Idée : On trie les arêtes par ordre croissant de poids, et on les ajoute à T , suivant leur ordre, tant qu'on ne crée pas de cycle.

Algorithme de Kruskal :

- **Entrée :** Un graphe pondéré G .
- **Initialisation :** T contient tous les sommets de G , mais aucune arête.
- **Traitement :**
 - On choisit une arête e de G qui n'est pas dans T et qui a le plus petit poids.
 - Si e ne crée pas de cycle dans T , on l'ajoute à T .
 - Sinon, on passe à l'arête suivante.
 - On s'arrête lorsque l'on a essayé toutes les arêtes de G .
- **Sortie :** L'arbre couvrant T de G .

Exemple

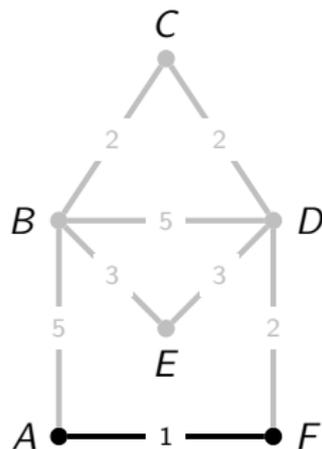
Ordre : AF (1) – CB, CD, DF (2) – BE, DE (3) – BD, BA (5)



Poids de T =

Exemple

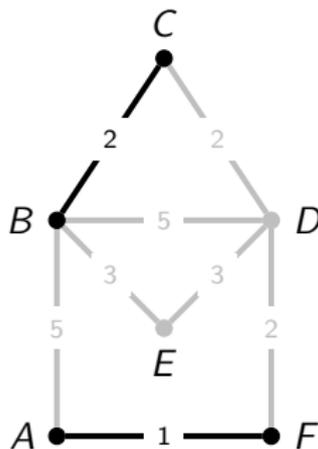
Ordre : AF (1) – CB, CD, DF (2) – BE, DE (3) – BD, BA (5)



Poids de $T = 1$

Exemple

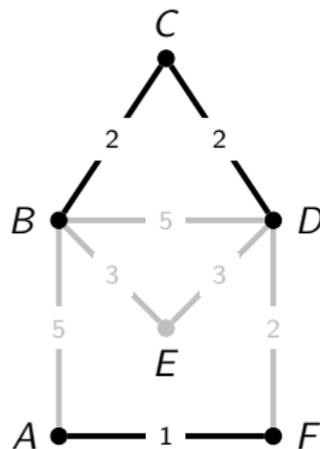
Ordre : AF (1) – CB, CD, DF (2) – BE, DE (3) – BD, BA (5)



Poids de $T = 1 + 2$

Exemple

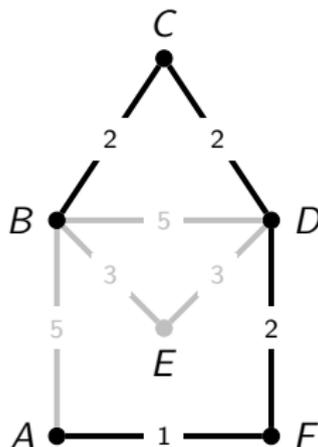
Ordre : AF (1) – CB , CD , DF (2) – BE , DE (3) – BD , BA (5)



Poids de $T = 1 + 2 + 2$

Exemple

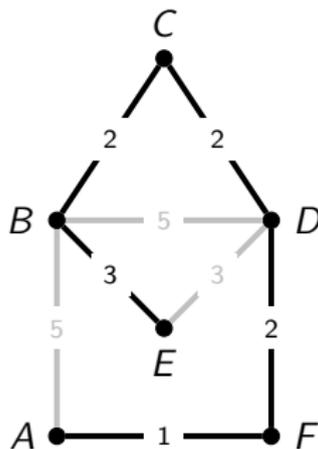
Ordre : AF (1) – CB, CD, DF (2) – BE, DE (3) – BD, BA (5)



$$\text{Poids de } T = 1 + 2 + 2 + 2$$

Exemple

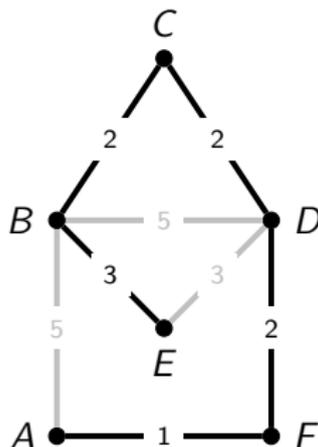
Ordre : AF (1) – CB, CD, DF (2) – BE, DE (3) – BD, BA (5)



$$\text{Poids de } T = 1 + 2 + 2 + 2 + 3$$

Exemple

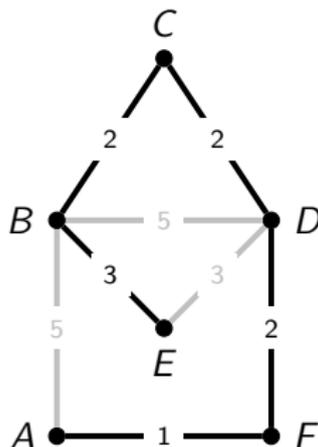
Ordre : AF (1) – CB, CD, DF (2) – BE, DE (3) – BD, BA (5)



$$\text{Poids de } T = 1 + 2 + 2 + 2 + 3$$

Exemple

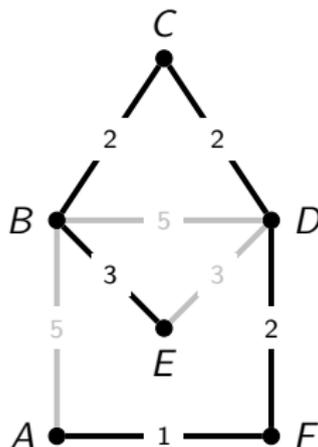
Ordre : AF (1) – CB, CD, DF (2) – BE, DE (3) – BD, BA (5)



$$\text{Poids de } T = 1 + 2 + 2 + 2 + 3$$

Exemple

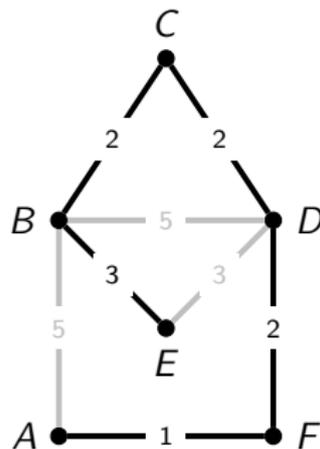
Ordre : AF (1) – CB, CD, DF (2) – BE, DE (3) – BD, BA (5)



$$\text{Poids de } T = 1 + 2 + 2 + 2 + 3$$

Exemple

Ordre : AF (1) – CB, CD, DF (2) – BE, DE (3) – BD, BA (5)



Poids de $T = 1 + 2 + 2 + 2 + 3 = 10$

2. Ordonnancement, méthode MPM

Contexte : Un projet, consistant en la réalisation de plusieurs tâches successives. On connaît, pour chaque tâche :

- sa durée ;
- sa dépendance aux autres tâches (antériorité).

Contexte : Un projet, consistant en la réalisation de plusieurs tâches successives. On connaît, pour chaque tâche :

- sa durée ;
- sa dépendance aux autres tâches (antériorité).

La **méthode MPM** permet de déterminer :

- 1 la date optimale de fin de projet ;
- 2 pour chaque tâche, les dates au plus tôt et au plus tard de réalisation ;
- 3 pour chaque tâche, les marges dont on dispose.

Exemple

Construire un voilier en minimisant le nombre de jours. Différentes étapes :

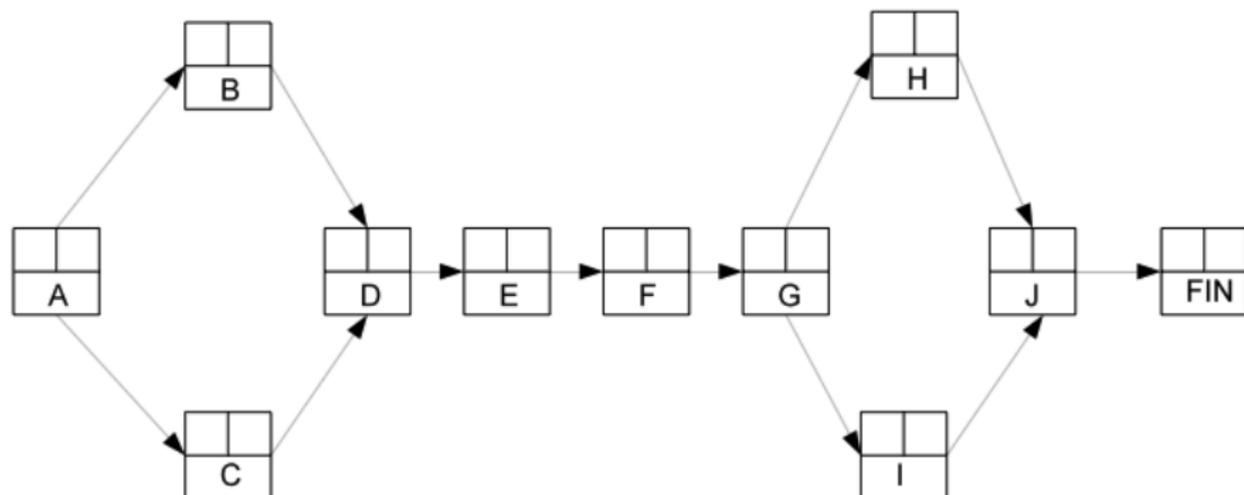
Code	Tâche	Durée en jours	Antériorité
<i>A</i>	Tracé des plans	3	–
<i>B</i>	Fabrication de la coque	9	<i>A</i>
<i>C</i>	Équipements techniques	4	<i>A</i>
<i>D</i>	Premiers tests	1	<i>B, C</i>
<i>E</i>	Montage des voiles	5	<i>D</i>
<i>F</i>	Seconds tests	1	<i>E</i>
<i>G</i>	Aménagement de la cabine	2	<i>F</i>
<i>H</i>	Réglages	1	<i>G</i>
<i>I</i>	Peinture, finitions	3	<i>G</i>
<i>J</i>	Derniers tests et mise à l'eau	1	<i>H, I</i>

Niveaux de dépendance

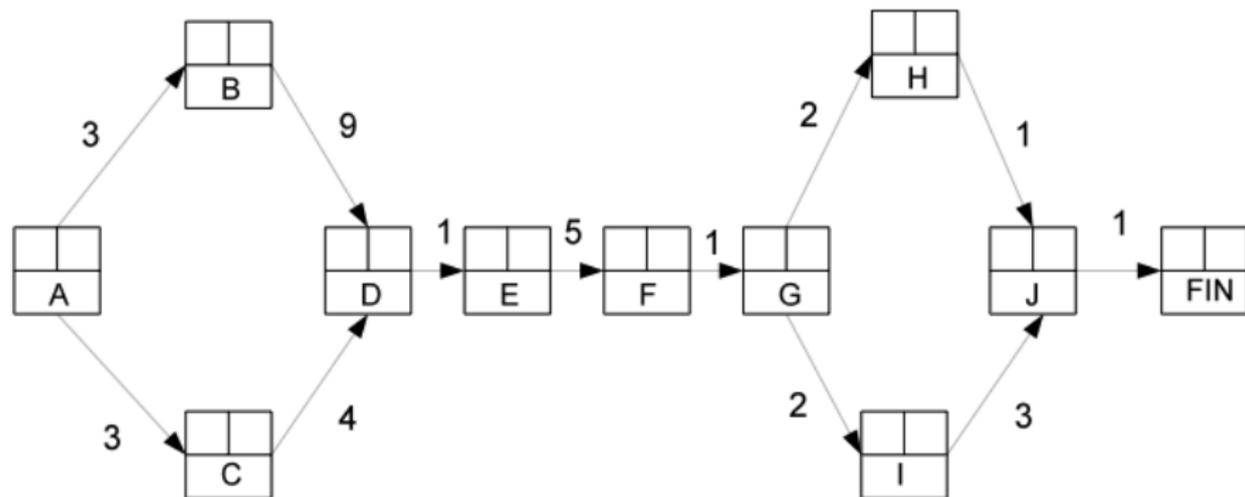
Étape 1 : Compléter les niveaux de dépendance.

Code	Tâche	Durée en jours	Antériorité	Niveau
<i>A</i>	Tracé des plans	3	–	0
<i>B</i>	Fabrication de la coque	9	<i>A</i>	1
<i>C</i>	Équipements techniques	4	<i>A</i>	1
<i>D</i>	Premiers tests	1	<i>B, C</i>	2
<i>E</i>	Montage des voiles	5	<i>D</i>	3
<i>F</i>	Seconds tests	1	<i>E</i>	4
<i>G</i>	Aménagement de la cabine	2	<i>F</i>	5
<i>H</i>	Réglages	1	<i>G</i>	6
<i>I</i>	Peinture, finitions	3	<i>G</i>	6
<i>J</i>	Derniers tests et mise à l'eau	1	<i>H, I</i>	7

Étape 2 : Création du graphe des tâches.

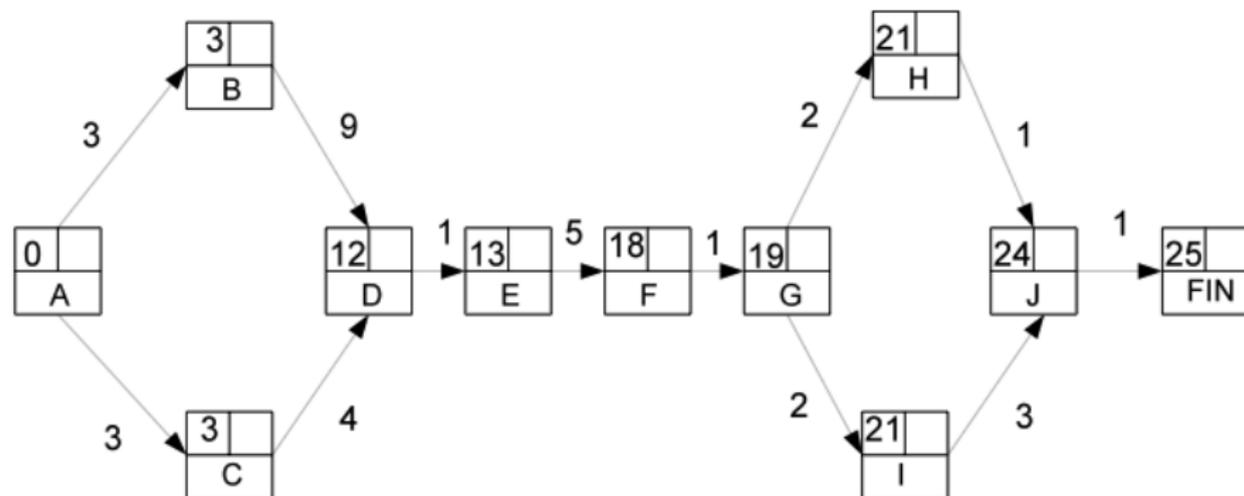


Étape 3 : Pondération des arcs.



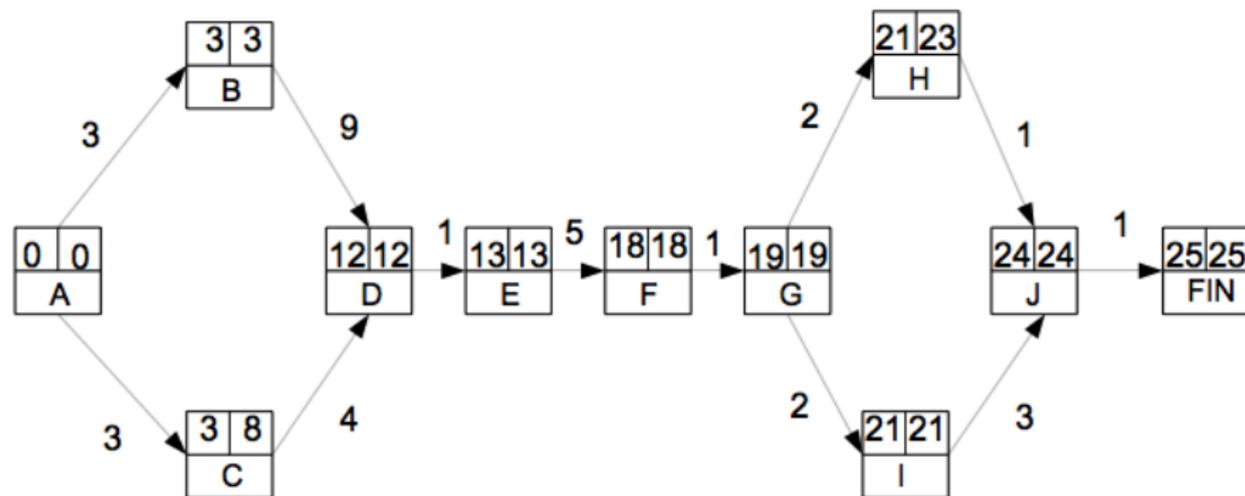
Méthode MPM

Étape 4 : Compléter les dates au plus tôt.



Méthode MPM

Étape 5 : Compléter les dates au plus tard.

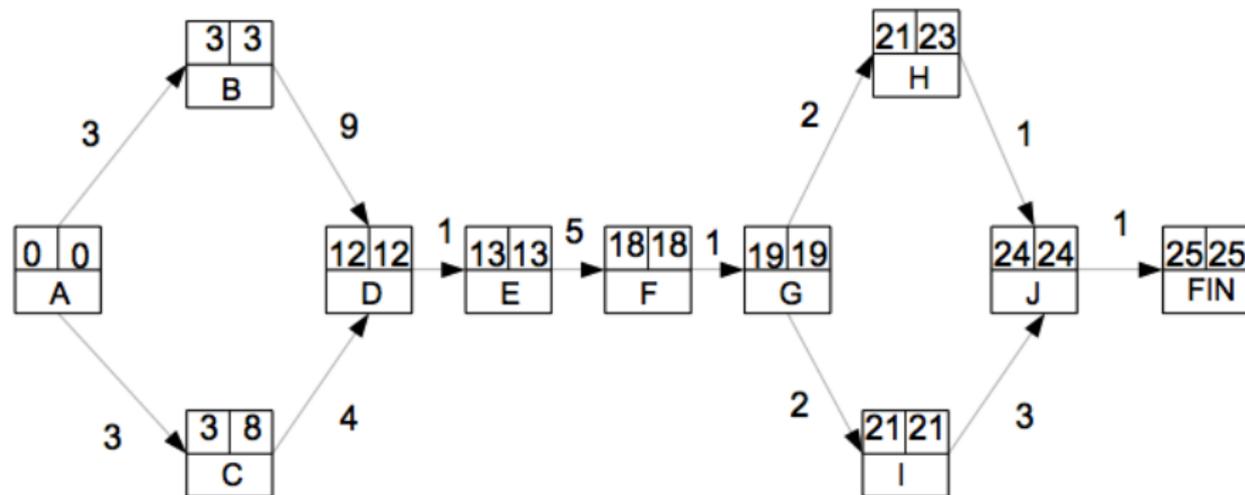


Tâches critiques

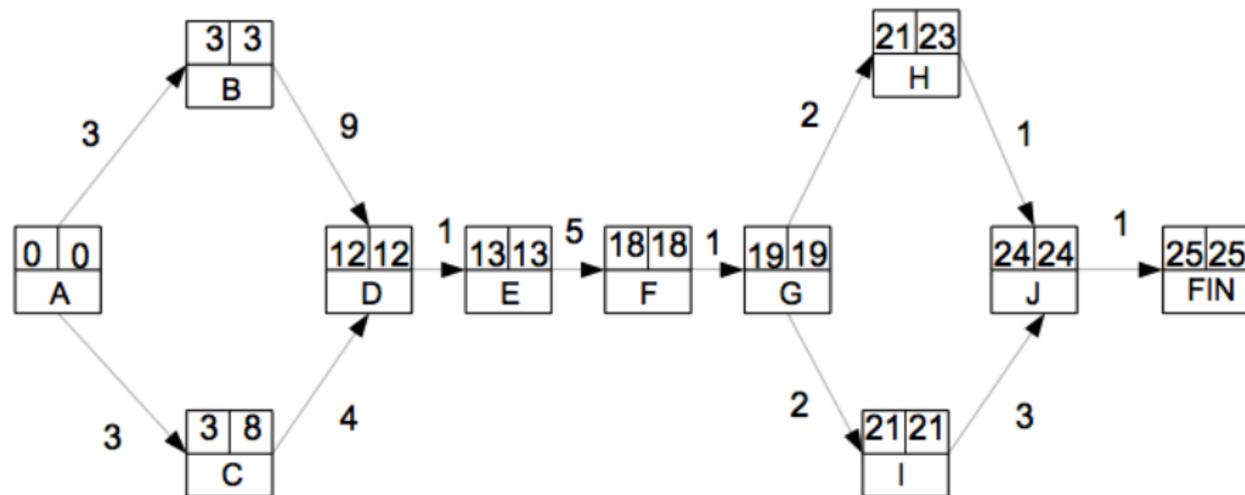
- **Tâche critique** = Date au plus tôt = Date au plus tard.
- **Chemin critique** = Un chemin de tâches critiques.

Un retard sur une tâche critique entraîne un retard sur la fin du projet !

Dans notre cas...



Dans notre cas...



Tâches critiques : A, B, D, E, F, G, I, J.

Chemin critique : A - B - D - E - F - G - I - J.

Marges

- **Totale** = Retard n'occasionnant pas de retard général.
- **Libre** = Retard ne remettant pas en cause les dates au plus tôt suivantes.

Marges

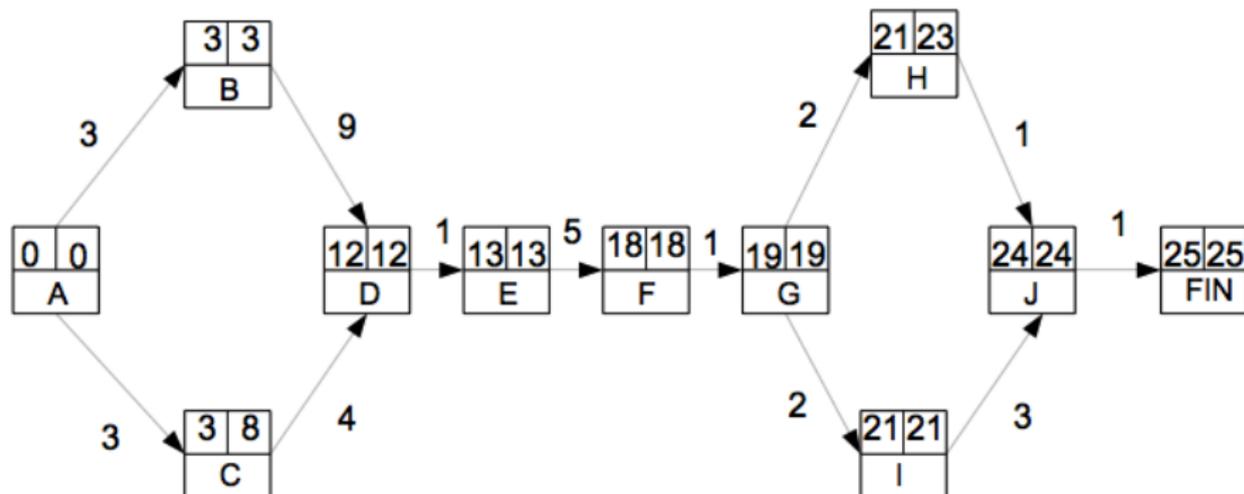
- **Totale** = Retard n'occasionnant pas de retard général.
- **Libre** = Retard ne remettant pas en cause les dates au plus tôt suivantes.

Remarques :

- Marge totale = Date au plus tard – Date au plus tôt.
- Marge libre = Date au plus tôt suivante – Date au plus tôt – Durée tâche.

Dans notre cas...

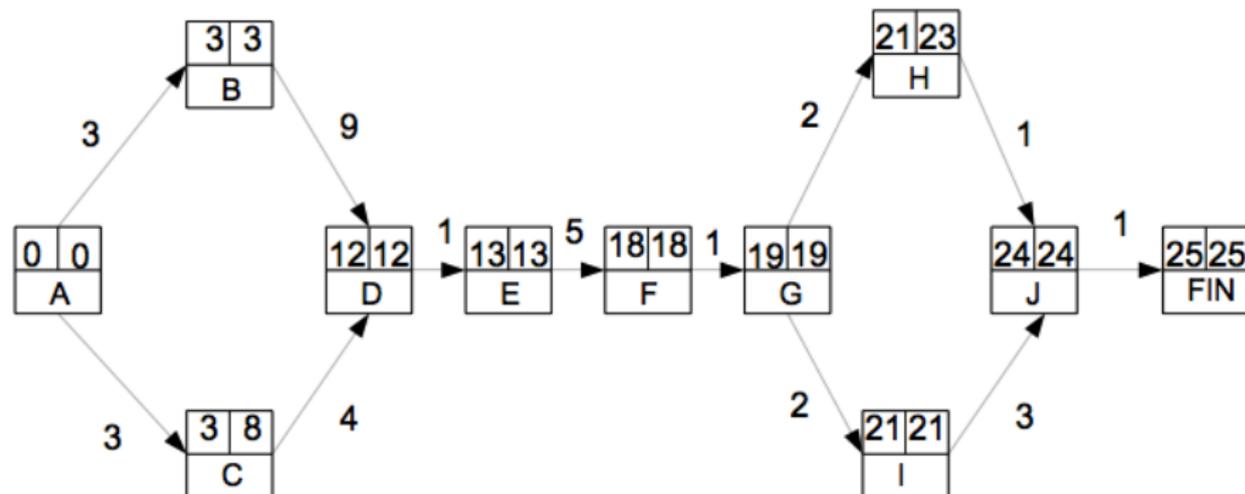
Étape 6 : Calcul des marges.



Tâche	A	B	C	D	E	F	G	H	I	J
Marge totale										
Marge libre										

Dans notre cas...

Étape 6 : Calcul des marges.



Tâche	A	B	C	D	E	F	G	H	I	J
Marge totale	$0 - 0 = 0$	$3 - 3 = 0$	$8 - 3 = 5$	0	0	0	0	2	0	0
Marge libre	$3 - 0 - 3 = 0$	$12 - 3 - 9 = 0$	$12 - 3 - 4 = 5$	0	0	0	0	$24 - 21 - 1 = 2$	0	0

- Durée minimale de construction : **25 jours**.
- Chemin critique à surveiller : **A – B – D – E – F – G – I – J**.
- De la marge sur : **C, H**.

Autre exemple

Un responsable de production est chargé de mettre en place le planning de réalisation d'un nouveau produit. Le tableau suivant résume les différentes tâches du projet :

Tâche	Durée en jours	Antériorité	Niveau
<i>A</i>	4,5	–	?
<i>B</i>	6,5	<i>A</i>	?
<i>C</i>	10	–	?
<i>D</i>	20	–	?
<i>E</i>	16	<i>C</i>	?
<i>F</i>	5	<i>C</i>	?
<i>G</i>	6	<i>D, F</i>	?
<i>H</i>	8	<i>B, G</i>	?
<i>I</i>	6	<i>H</i>	?
<i>J</i>	2	<i>E, H</i>	?

À faire

- 1 Compléter la colonne niveau.
- 2 Que signifie, pour une tâche, d'avoir un niveau nul ?
- 3 Construire le diagramme MPM du projet.
- 4 Déduire le chemin critique.
- 5 Déduire les marges totales et libres de chaque tâche.

Étape 1 : Compléter les niveaux de dépendance.

Tâche	Durée en jours	Antériorité	Niveau
<i>A</i>	4,5	–	0
<i>B</i>	6,5	<i>A</i>	1
<i>C</i>	10	–	0
<i>D</i>	20	–	0
<i>E</i>	16	<i>C</i>	1
<i>F</i>	5	<i>C</i>	1
<i>G</i>	6	<i>D, F</i>	2
<i>H</i>	8	<i>B, G</i>	3
<i>I</i>	6	<i>H</i>	4
<i>J</i>	2	<i>E, H</i>	4

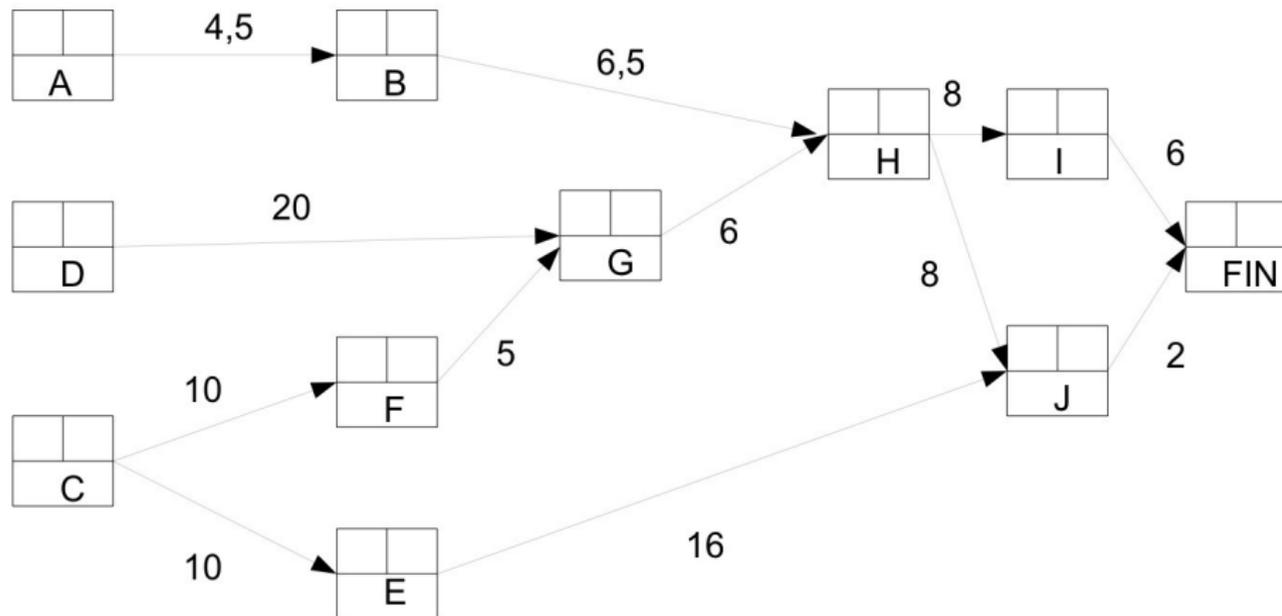
Étape 1 : Compléter les niveaux de dépendance.

Tâche	Durée en jours	Antériorité	Niveau
<i>A</i>	4,5	–	0
<i>B</i>	6,5	<i>A</i>	1
<i>C</i>	10	–	0
<i>D</i>	20	–	0
<i>E</i>	16	<i>C</i>	1
<i>F</i>	5	<i>C</i>	1
<i>G</i>	6	<i>D, F</i>	2
<i>H</i>	8	<i>B, G</i>	3
<i>I</i>	6	<i>H</i>	4
<i>J</i>	2	<i>E, H</i>	4

Antériorité nulle = Pas de prérequis pour la tâche.

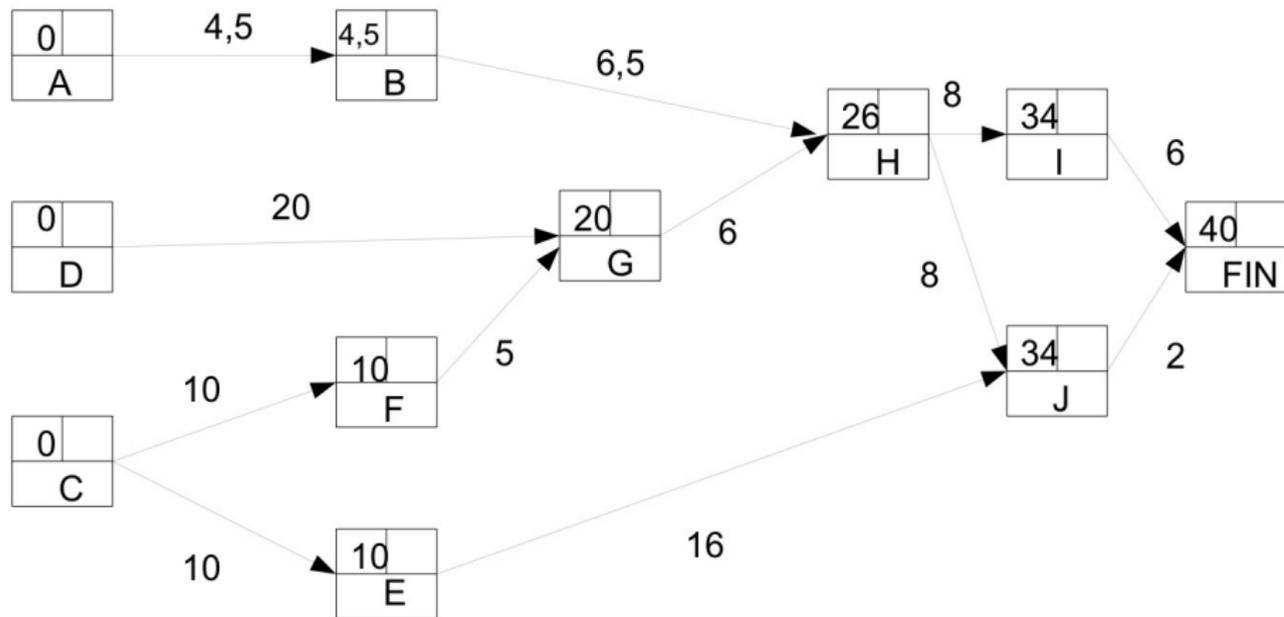
Solution

Étapes 2 & 3 : Création du graphe pondéré des tâches.



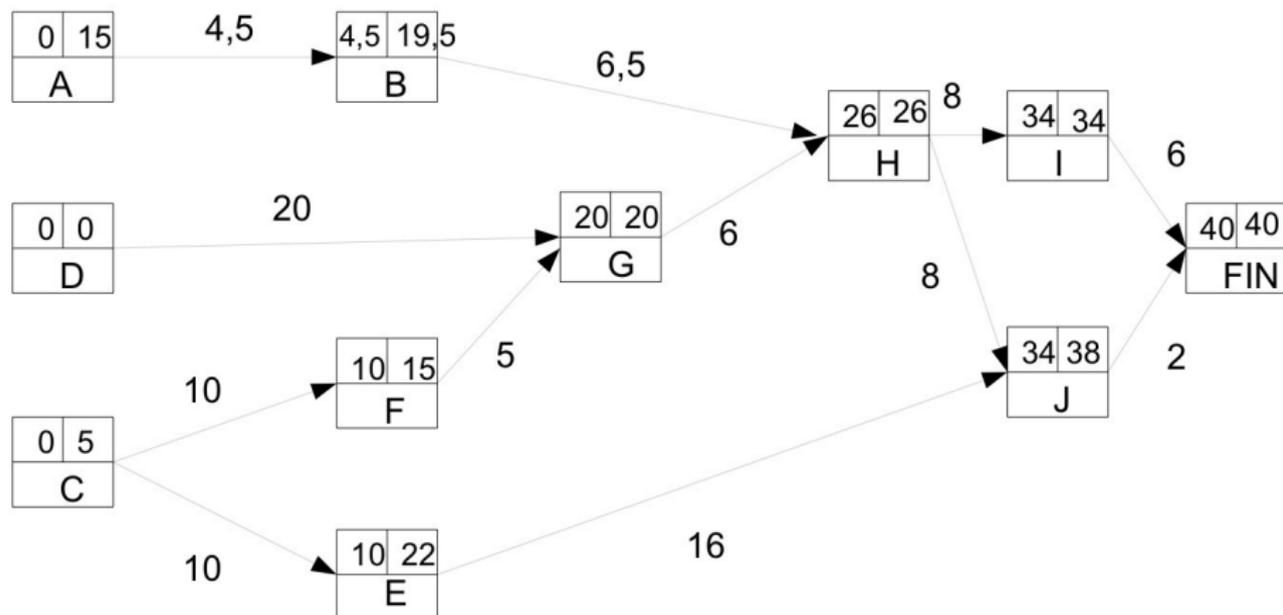
Solution

Étape 4 : Compléter les dates au plus tôt.



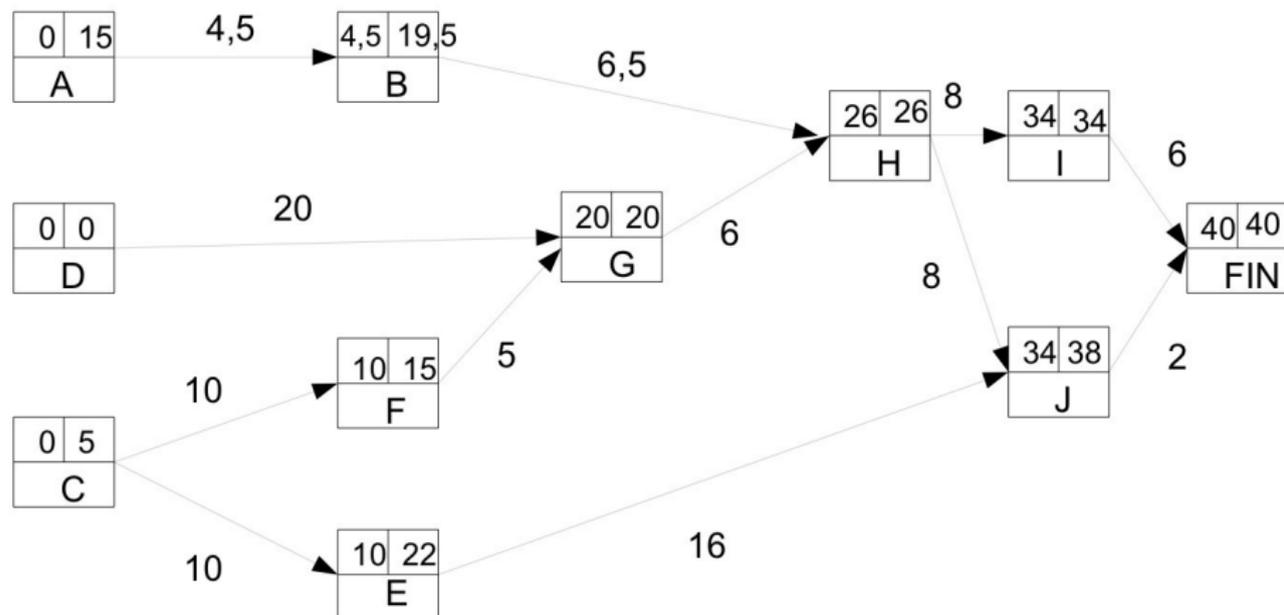
Solution

Étape 5 : Compléter les dates au plus tard.



Solution

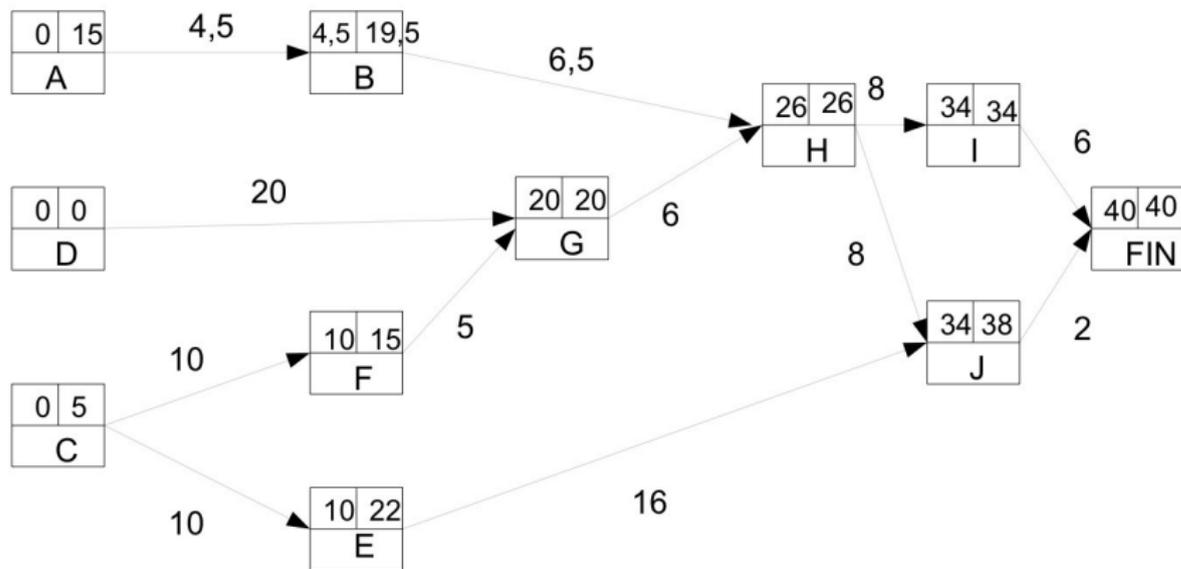
Étape 5 : Compléter les dates au plus tard.



Chemin critique : **D – G – H – I.**

Solution

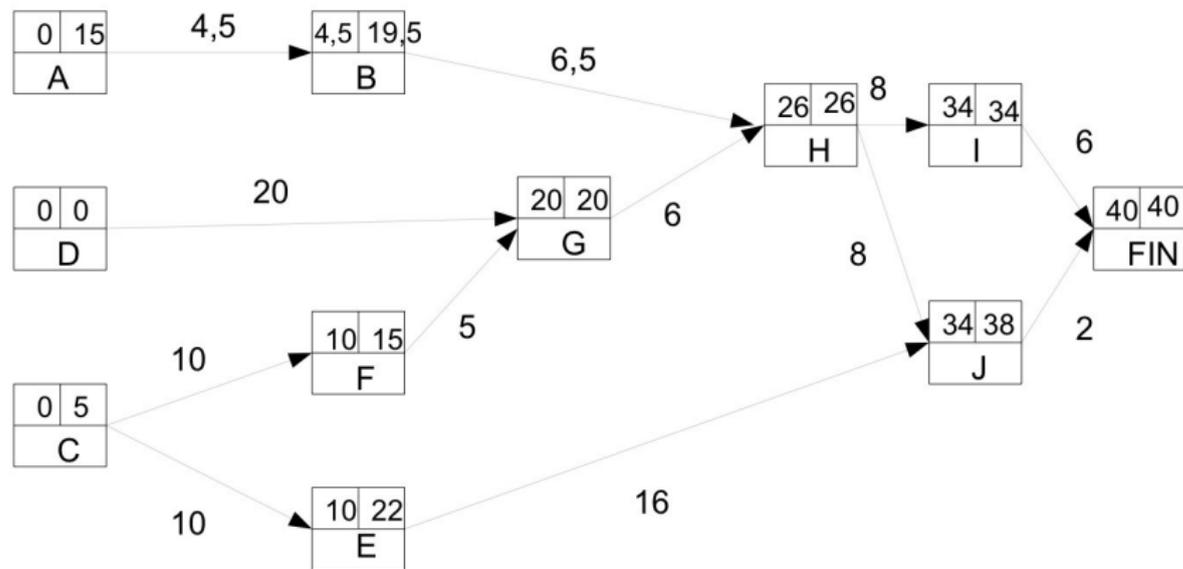
Étape 6 : Calcul des marges.



Tâche	A	B	C	D	E	F	G	H	I	J
Marge totale										
Marge libre										

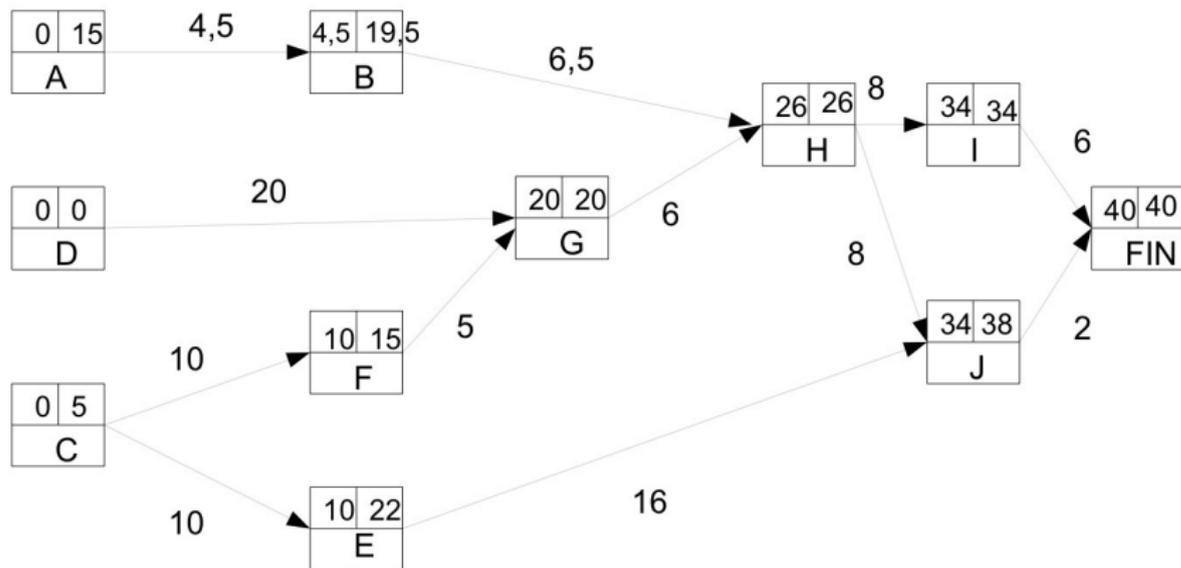
Solution

Étape 6 : Calcul des marges.



Tâche	A	B	C	D	E	F	G	H	I	J
Marge totale	15	15	5	0	12	5	0	0	0	4
Marge libre	0	15	0	0	8	5	0	0	0	4

Solution : marge totale VS marge libre



- On peut retarder la tâche *E* de 12 jours sans retarder le projet.
- Si on retarde *E* de plus de 8 jours, cela retardera la tâche *J*.