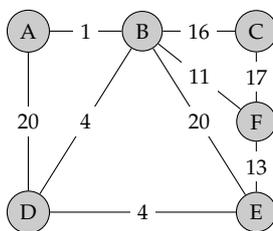


## Optimisation

**Exercice 1 (Arbres couvrants)** On considère un réseau de 5 villes. Le coût de la construction d'une ligne électrique entre  $i$  et  $j$  est  $A_{ij}$  avec  $A$  donnée par :

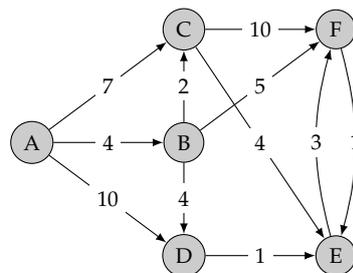
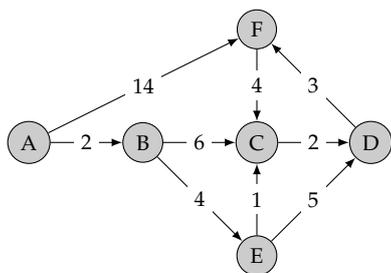
$$A = \begin{bmatrix} 0 & 3 & 5 & 11 & 9 \\ 3 & 0 & 3 & 9 & 8 \\ 5 & 3 & 0 & +\infty & 10 \\ 11 & 9 & +\infty & 0 & 7 \\ 9 & 8 & 10 & 7 & 0 \end{bmatrix}$$

1. Tracer le graphe valué associé.
2. Trouvez le coût minimum d'un réseau liant les villes entre elles à l'aide de l'algorithme de Kruskal. Combien d'itérations sont t'elles nécessaire ?
3. Répondre à la même question en utilisant l'algorithme de Prim pour le graphe suivant :

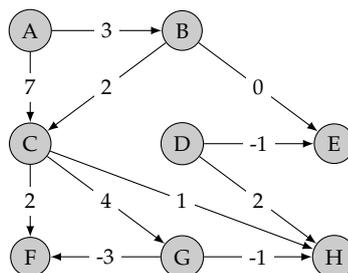
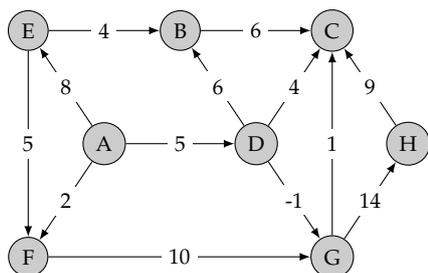


4. Peut on adapté les deux algorithmes pour trouver un arbre couvrant de coût maximal? Appliquez cet algorithme à l'un des graphes précédents.

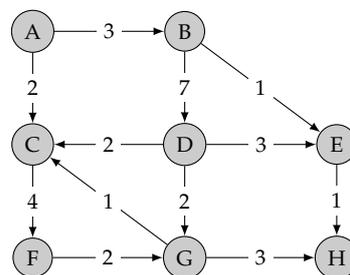
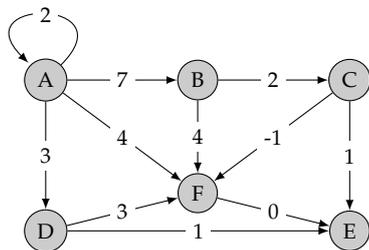
**Exercice 2 (Algorithme de Dijkstra)** Appliquer l'algorithme de Dijkstra aux graphes suivant pour calculer les chemins de poids minimum depuis le sommet A. Combien d'itérations sont réalisés ?



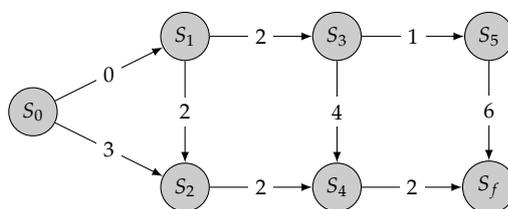
**Exercice 3 (Algorithme de Bellman simplifié)** Décomposer les graphes suivants en niveaux puis appliquer l'algorithme de Bellman simplifié pour calculer les chemins de poids minimum et maximum depuis A. Combien d'itérations sont réalisés ?



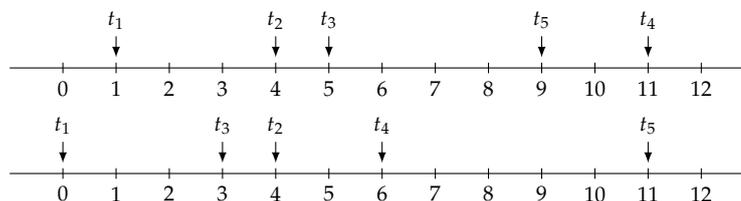
**Exercice 4 (Algorithme de Bellman-Ford)** Appliquer l'algorithme de Bellman-Ford pour calculer les distances minimale depuis le sommet A aux autres sommets. Dans chaque cas, quel autre algorithme aurait on pu utiliser ? Pourquoi ?



**Exercice 5** On considère un projet composé de 5 tâches  $(S_i)_{i=1,\dots,5}$  plus deux tâches fictives,  $S_0$  et  $S_f$ , de début et de fin de projet. Ces projets sont représentés par des graphes potentiel-tâches et on note  $t_i$  le début de la tâche  $S_i$  (et  $t_0 = 0$ ).



1. Lequel des ordonnancements suivants n'est pas compatible avec ce graphe ?



- Calculer l'ordonnancement au plus tôt du projet et sa durée minimale.
- Donner l'ordonnancement au plus tard pour terminer le projet en 10 jours.

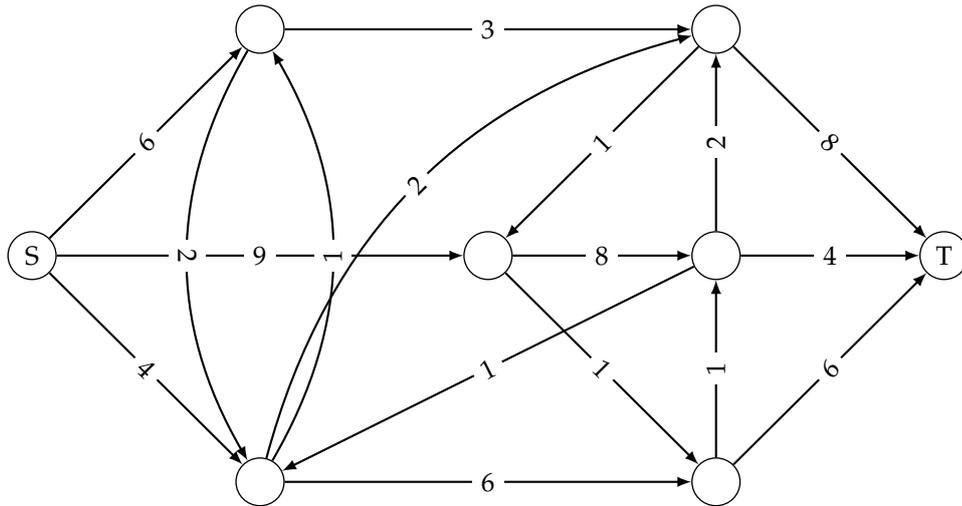
**Exercice 6** Le prince est parti à la recherche du trésor ; il peut accomplir les actions suivantes :

- Du point de départ, aller à la ville du marché, en contournant la rivière par un gué : 4 jours.
- Du point de départ, traverser la forêt : 1 jour.
- Depuis la forêt, abattre des arbres pour traverser la rivière, et se rendre à la ville du marché : 2 jours.
- Depuis la forêt, se rendre à la capitale provinciale en traversant les marais : 7 jours.
- S'équiper chaudement au marché, et partir pour le col du nord : 5 jours.
- Trouver un bon cheval au marché, et se rendre à la capitale provinciale par la grand-route : 3 jours.
- Depuis le col du nord, se rendre au refuge du devin : 3 jours.
- Depuis la capitale provinciale, se rendre au refuge du devin : 4 jours.
- Se rendre de la capitale provinciale au palais du roi, en étant retardé par des contrôles : 10 jours.
- Au sortir du devin, partir directement chercher l'épée, et la trouver après s'être perdu par manque de carte : 20 jours.
- Au sortir de chez le devin, au mépris de ses avis, se rendre directement à la grotte et tuer le dragon avec un canif : 32 jours (il faut du temps pour le tuer avec un canif).
- Bien conseillé par le devin, prendre un raccourci pour le palais du roi : 5 jours.
- Un fois arrivé au palais du roi, s'éduire la bibliothécaire, puis trouver les cartes qui expliquent l'emplacement de l'épée et du trésor : 6 jours.

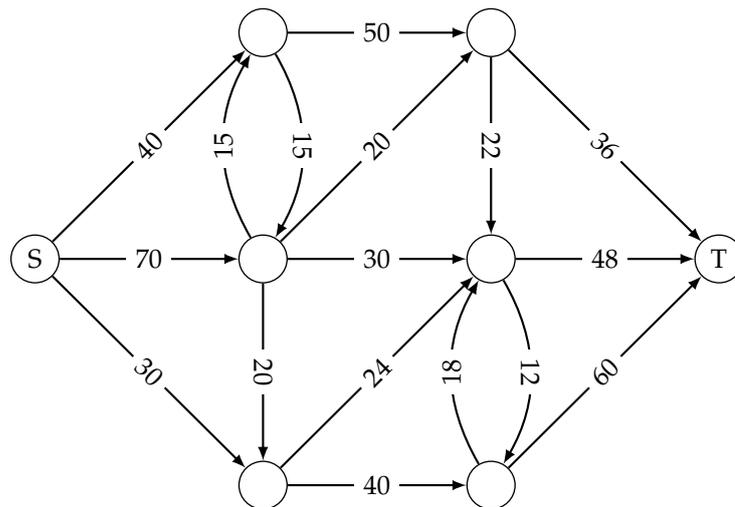
- En utilisant les cartes trouvées dans la bibliothèque, faire tout le tour de la montagne, et traverser un labyrinthe qui mène directement au trésor : 30 jours.
- En utilisant les cartes, aller chercher l'épée pour combattre le dragon : 7 jours.
- S'entraîner à l'épée, puis tuer le dragon : 8 jours.
- Une fois l'épée trouvée, au lieu d'affronter le dragon, utiliser l'épée pour creuser un tunnel par dessous, et déboucher directement dans la cachette du trésor : 18 jours.
- Une fois le dragon tué, résoudre l'énigme qui ouvre la cachette du trésor : 9 jours.

Comment doit-il faire pour récupérer le trésor le plus vite possible ? Quel temps lui faudra-t-il ? Pour répondre à ces questions, modéliser le problème à l'aide d'un graphe et utiliser un algorithme vu en cours.

**Exercice 7 (Flot maximum)** Déterminez le flot maximum pour le réseau ci-dessous. Comment montrer que la solution proposée est bien optimale ?

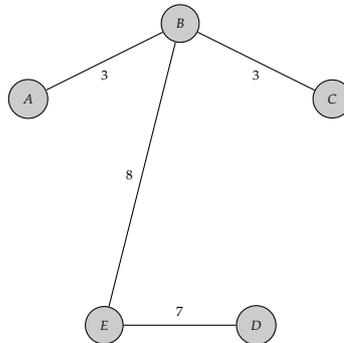


**Exercice 8 (Mon patron a tort)** Soit le réseau représenté plus bas. Votre patron est convaincu qu'il est possible de faire passer 138 unités de flots de  $s$  à  $t$  et il vous reproche de ne pas être capable d'exhiber un tel flot. Convincez votre patron par un argument simple qu'un tel flot n'existe pas. Quelle est la valeur maximale d'un flot dans ce réseau ?



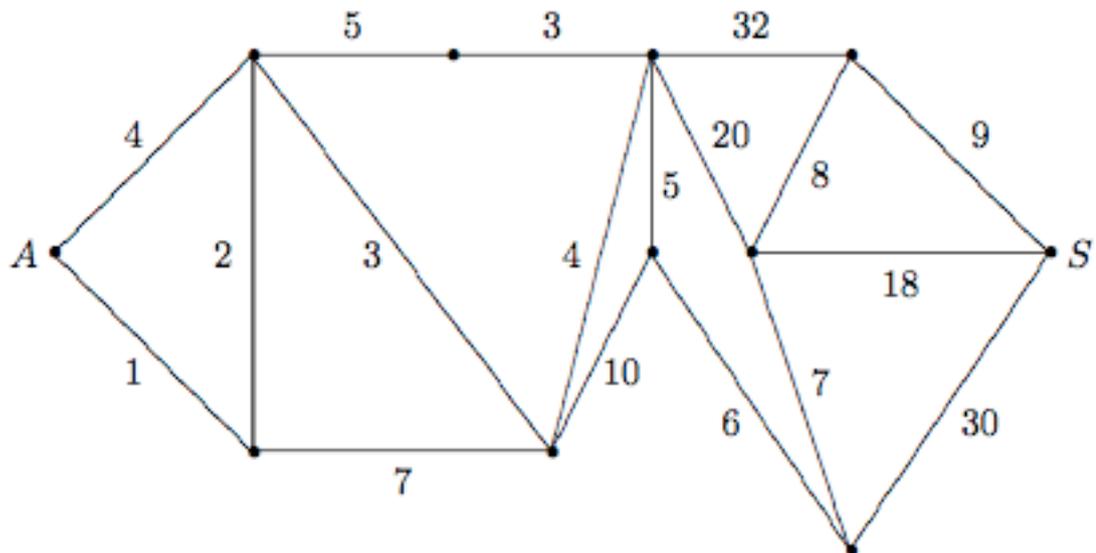
## Optimisation (Solutions)

**Correction 1** Nous réalisons  $m$  itérations de Kruskal avec  $m = n - 1 = 4$ . Nous obtenons :



Nous obtenons, par conséquent, un arbre de poids 21.

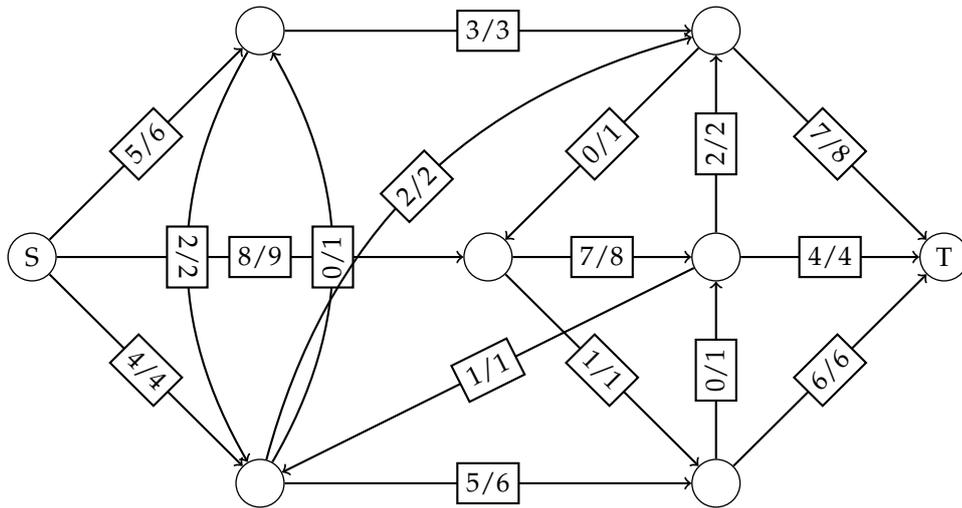
**Correction 6**



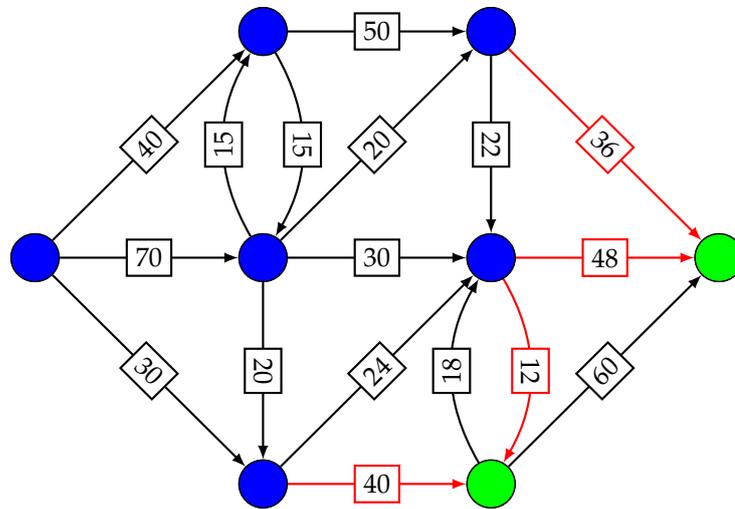
**Correction 7** On voit que  $f_{\text{net}}(S) = -f_{\text{net}}(T) = 17$ . Si on essaie de trouver un chemin  $f$ -augmentant avec un BFS ou DFS en s'autorisant donc à prendre,

- Soit les arêtes non  $f$ -saturées,
- Soit les arêtes telles qu'il y ait une arête dans l'autre sens non  $f$ -nulles (back edges),

on part de  $S$  mais on n'arrive jamais à  $T$ . On a donc  $f_{\text{max}} = 17$ .



**Correction 8** On sait par la dualité que pour toute coupe  $S \rightarrow \bar{S}$ ,  $f_{max} \leq \text{capacité}(S \rightarrow \bar{S})$ . En prenant dans  $S$  les noeuds bleus et dans  $\bar{S}$  les noeuds verts, capacité( $S \rightarrow \bar{S}$ ) vaut la somme des capacités des arêtes rouges, c'est-à-dire 136. On a donc  $f_{max} \leq 136$ .



Soit  $f$  le flot du graph suivant, valeur( $f$ ) = 136. On sait donc que  $136 = \text{valeur}(f) \leq f_{max} \leq 136$ . On a donc nécessairement  $f_{max} = 136$ .

