

Examen

Durée 2h

Exercice 1. On considère le graphe G représenté par la figure suivante :

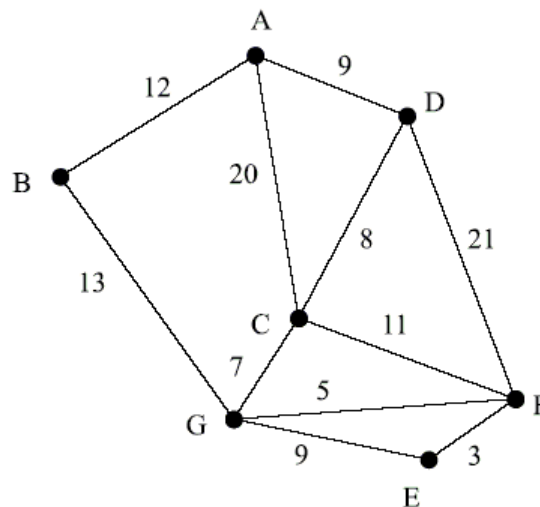


FIGURE 1 – Graphe G

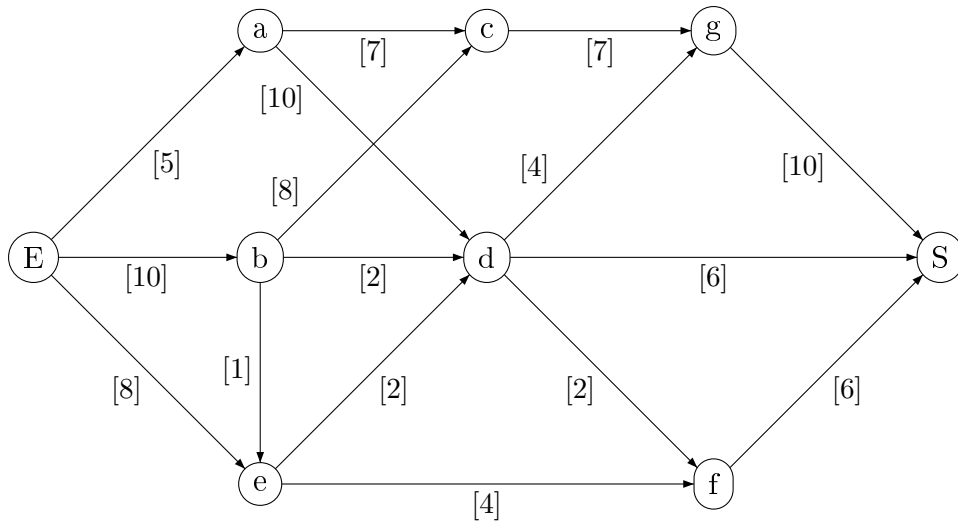
Appliquer l'algorithme de Floyd-Warshall pour obtenir les plus courts chemins entre toute paire de nœuds du graphe.

Exercice 2.

Avant d'établir un projet de construction d'autoroute, on désire étudier la capacité du réseau routier, représenté par le graphe ci-dessous reliant la ville E à la ville S . Pour cela, on a évalué le nombre maximal de véhicules que chaque route peut écouler par heure, compte tenu des ralentissements aux traversées des villes et villages, des arrêts aux feux, etc. Ces évaluations sont indiquées en centaines de véhicules par heure sur les arcs du graphe. Les temps de parcours entre villes sont tels que les automobilistes n'emprunteront que les chemins représentés par le graphe.

En utilisant l'algorithme de Ford-Fulkerson, calculer le flot maximal pouvant circuler de E à S .

Donner la coupe minimale associée au flot maximal.



Exercice 3. On s'intéresse au problème de la recherche d'une plus longue sous-séquence commune à deux mots. Etant donné un mot ω , on appelle sous-séquence de ω un mot $\omega_{j_1}\omega_{j_2}\dots\omega_{j_k}$ avec $j_1 < j_2 < \dots < j_k$. Notons $s(\omega, \omega')$ la plus longue sous-séquence commune à ω et ω' , c'est-à-dire le mot le plus long qui soit une sous-séquence de ω et une sous-séquence de ω' .

1. Que valent $s(\omega, \epsilon)$ et $s(\epsilon, \omega)$?
2. On considère deux mots ω et ω' , et deux caractères α et β (non-nécessairement distincts). Exprimer $s(\omega\alpha, \omega'\beta)$ en fonction de $s(\omega\alpha, \omega')$, $s(\omega, \omega'\beta)$ et $s(\omega, \omega')$, en distinguant les cas $\alpha = \beta$ et $\alpha \neq \beta$.
3. Dédire en un algorithme calculant la plus longue sous-séquence commune à deux mots : cet algorithme consistera à remplir un tableau dont les lignes et les colonnes sont indexés par les caractères des deux mots, et dont la case (i, j) contiendra la longueur de la plus longue sous-séquence commune aux préfixes de longueurs i et j des deux mots.
4. Appliquer cet algorithme aux mots BDCABA et ABCBDAB.
5. Adapter cet algorithme pour qu'il calcule le *diff* entre deux textes.