Licence Informatique — Algorithmique 2 — Épreuve de septembre 2004
Durée trois heures — Documents et calculatrices interdits

---

1 Arbres couvrants de valeur minimale (6 points)

Le professeur Machin propose l'algorithme suivant pour calculer un arbre couvrant de valeur minimale d'un graphe $G = (S, A)$. Le graphe G est supposé non orienté et connexe. La fonction est censée retourner l'ensemble des arêtes de l'arbre.

function arbre_couvrant_minimal (S, A)
begin
  if S contient un ou deux sommets then
    return A
  else
    partitionner S en deux sous–ensembles S_1 et S_2 de même cardinal ± 1
    soit A_1 l'ensemble des arêtes de A qui ne relient que des sommets de S_1
    soit A_2 l'ensemble des arêtes de A qui ne relient que des sommets de S_2
    note : on s'est arrangé lors du partitionnement pour que les graphes (S_1, A_1) et (S_2, A_2) soient connexes
    T_1 := arbre_couvrant_minimal (S_1, A_1)
    T_2 := arbre_couvrant_minimal (S_2, A_2)
    a := une arête de valeur minimale parmi celles qui n'appartiennent ni à A_1 ni à A_2
    return $T_1 \cup T_2 \cup \{a\}$
  end

Question 1 [3 pts]. Montrer que la fonction retourne bien un arbre (raisonner par récurrence sur le cardinal de A ; utiliser un théorème vu en cours).

SOLUTION. On raisonne par récurrence sur le nombre n de sommets du graphe.
Base de la récurrence. Le graphe est connexe. Si $n < 2$ alors le graphe est sans cycle. C'est donc un arbre par définition.
Le cas général. On suppose $n \geq 2$. Hypothèse de récurrence : l'algorithme retourne un arbre si on l'applique à un graphe connexe de moins de n sommets. Notons n_1 et n_2 les nombres de sommets de S_1 et de S_2. On a $n_1 + n_2 = n$ et $n_1, n_2 < n$. Les graphes (S_1, A_1) et (S_2, A_2) sont connexes. L'hypothèse de récurrence s'applique : les deux appels récursifs retournent des arbres et on a $|T_1| = n_1 - 1$, $|T_2| = n_2 - 1$ et donc $|T_1 \cup T_2| = n_1 + n_2 - 2 = n - 2$. L'arête a connecte T_1 et T_2 donc $T_1 \cup T_2 \cup \{a\}$ est connexe. Ce graphe comporte n sommets et $n - 2 + 1 = n - 1$ arêtes. C'est donc un arbre d'après un théorème du cours (point 2). CQFD.

Question 2 [3 pts]. Montrer que l'arbre couvrant retourné par la fonction est de valeur minimale ou montrer que l'algorithme est faux en exhibant un contre–exemple.

SOLUTION. L'algorithme est faux. Considérer par exemple un triangle dont une arête vaut 2 et les deux autres 1.

2 Graphes eulériens (4 points)

On rappelle qu'un graphe non orienté est *eulérien* si tous ses sommets sont de degré pair sauf éventuellement deux (le degré d'un sommet x étant le nombre d'arêtes ayant x pour extrémité).

Question 3 [1 pt]. À quel célèbre problème historique mentionné en cours la notion de graphe eulérien s'applique–t–elle ? Quel mathématicien a résolu ce problème ?

> SOLUTION. Les ponts de Königsberg. Leonhard Euler.

La fonction suivante détermine si un graphe non orienté G est eulérien. On suppose que le graphe comporte n sommets, m arêtes et qu'il est représenté par des listes d'adjacence.

```
function graphe_eulérien_? (G)
begin
   nombre_de_sommets_de_degré_impair := 0
   for tout sommet x de G do
      degré := 0
      for tout voisin de x do
         degré := degré + 1
      od
      if degré est impair then
         nombre_de_sommets_de_degré_impair := nombre_de_sommets_de_degré_impair + 1
      fi
   od
   return nombre_de_sommets_de_degré_impair vaut 0 ou 2
end
```

Question 4 [2 pts]. Donner pour chaque affectation une estimation asymptotique (notation « O ») du nombre total de fois où elle est exécutée.

> SOLUTION. La première affectation est effectuée 1 et donc $O(1)$ fois. La deuxième est effectuée n et donc $O(n)$ fois. La troisième est effectuée $2m$ et donc $O(m)$ fois. La quatrième est effectuée au plus n fois et donc $O(n)$ fois.

Question 5 [1 pt]. En déduire la complexité en temps dans le pire des cas de la fonction.

> SOLUTION. La complexité en temps dans le pire des cas de la fonction est $O(n+m)$.

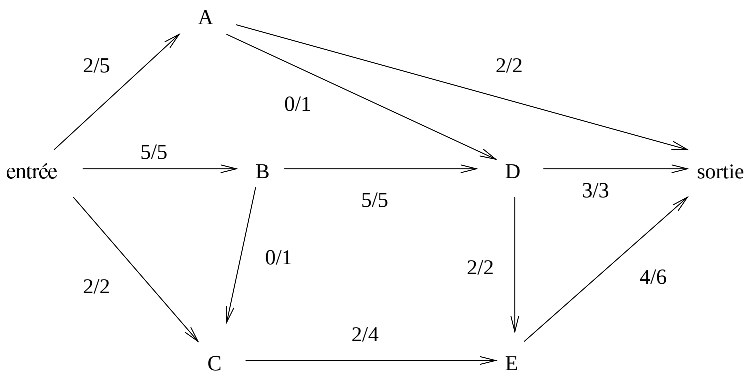
3 Ford–Fulkerson (3 points)

Question 6 [2 pts]. Appliquer l'algorithme de Ford–Fulkerson au flot suivant (donner à chaque itération une chaîne améliorante, mais ne pas détailler le calcul de la chaîne).

SOLUTION. On trouve une chaîne améliorante : $(\textit{entrée}, A)$, (A, D), (D, B), (B, C), (C,E) et $(E, \textit{sortie})$. Le long de cette chaîne, le flot peut être augmenté de 1. Faire le dessin. Le nouveau flot transitant par le réseau a pour valeur 10. Ensuite, on ne trouve plus de chaîne améliorante (le flot est maximal).

Question 7 [1 pt]. À la fin de l'exécution de l'algorithme, montrer que le flot est bien maximal en utilisant un théorème vu en cours.

SOLUTION. On utilise le théorème « flot maximal et coupe minimale » qui dit que le flot maximal a pour valeur la capacité minimale des coupes. Il suffit d'exhiber une coupe de capacité 10. C'est le cas de la coupe $\{\textit{entrée}, A\}$, $\{B, C, D, E, \textit{sortie}\}$.



4 Le simplexe (9 points)

Dans le but d'écouler deux kilos d'un ingrédient X proches de leur date de péremption, un restaurateur se décide à offrir une salade gratuite à ses clients. La salade est composée non seulement de l'ingrédient X mais aussi d'un autre ingrédient Y. Chacun des ingrédients coûte dix euros le kilo. La salade doit donc comporter au moins deux kilos de X (ceux qu'on veut écouler) et au plus trois kilos de Y (dont le stock est limité). Pour que la salade soit équilibrée, il faut aussi que la moyenne pondérée ci–dessous soit comprise entre 3 et 4 :

$$\frac{2\,x + 5\,y}{x + y}$$

où x et y désignent les nombres de kilos d'ingrédients X et Y qui entrent dans la composition de la salade. Le restaurateur souhaite limiter au maximum le coût de l'opération.

Question 8 [2 pts]. Modéliser le problème ci–dessus au moyen d'un programme linéaire (P). Bien préciser le sens des variables. Attention à la contrainte sur l'équilibre de la salade, présentée sous la forme d'une contrainte non linéaire dans l'énoncé mais qui peut se traduire sous la forme d'une ou deux contraintes linéaires.

SOLUTION. On note x et y les nombres de kilos d'ingrédients X et Y qui entrent dans la composition de la salade. On rend linéaire la contrainte qui ne l'est pas en multipliant les deux membres des inégalités par le dénominateur (comme il est positif, les inégalités ne changent pas de sens).

$$(P)\left\{\begin{array}{rcll} 10\,x+10\,y & = & z[\mathit{min}] & \\ x & \geq & 2 & \text{(stock à écouler)} \\ y & \leq & 3 & \text{(stock limité)} \\ x-2\,y & \leq & 0 & (\frac{2\,x+5\,y}{x+y}\geq 3) \\ -2\,x+y & \leq & 0 & (\frac{2\,x+5\,y}{x+y}\leq 4) \\ x,y & \geq & 0 & \end{array}\right.$$

Question 9 [2 pts]. Résoudre le problème graphiquement.

SOLUTION. On trouve $x = 2$, $y = 1$ et un objectif réalisé de 30. Faire le dessin bien sûr.

Question 10 [2 pts]. Écrire le programme dual (D) de (P).

SOLUTION. Il faut commencer par mettre (P) sous forme canonique. Au vu de la question suivante, on transforme (P) en un problème de minimisation avec des contraintes $\geq$.

$$(P)\left\{\begin{array}{rcll} 10\,x+10\,y & = & z[\mathit{min}] & \\ x & \geq & 2 & \text{(stock à écouler)} \\ -y & \geq & -3 & \text{(stock limité)} \\ -x+2\,y & \geq & 0 & (\frac{2\,x+5\,y}{x+y}\geq 3) \\ 2\,x-y & \geq & 0 & (\frac{2\,x+5\,y}{x+y}\leq 4) \\ x,y & \geq & 0 & \end{array}\right.$$

Le programme dual s'écrit alors

$$(D)\left\{\begin{array}{rcll} 2\,y_1-3\,y_2 & = & w[\mathit{max}] & \\ y_1-y_3+2\,y_4 & \leq & 10 & \text{(ingrédient } X) \\ -y_2+2\,y_3-y_4 & \leq & 10 & \text{(ingrédient } Y) \\ y_i & \geq & 0 & \end{array}\right.$$

Question 11 [2 pts]. Résoudre (D) par l'algorithme du tableau simplicial.

SOLUTION. On trouve $(y_1, \ldots, y_4) = (15, 0, 5, 0)$ et un objectif réalisé de 30 (égal comme il se doit à l'objectif réalisé à l'optimum par le primal).

```
          y1        y2        y3        y4        y5        y6
    ------------------------------------------------------------
 y5        1         0        -1         2         1         0 |         10
 y6        0        -1         2        -1         0         1 |         10
    ------------------------------------------------------------
           2        -3         0         0         0         0 |          0 = - w0

          y1        y2        y3        y4        y5        y6
    ------------------------------------------------------------
 y1        1         0        -1         2         1         0 |         10
 y6        0        -1         2        -1         0         1 |         10
    ------------------------------------------------------------
```

```
          0       -3        2       -4       -2        0 |       -20 = - w0

         y1       y2       y3       y4       y5       y6
   -----------------------------------------------------------
y1        1     -1/2        0      3/2        1      1/2 |        15
y3        0     -1/2        1     -1/2        0      1/2 |         5
   -----------------------------------------------------------
          0       -2        0       -3       -2       -1 |       -30 = - w0
```

Question 12 [1 pt]. Déduire de la réponse à la question précédente la solution optimale de (P).

SOLUTION. Les variables d'écart y_5 et y_6 correspondent respectivement à x et à y. On a donc à l'optimum $x = -f_5 = 2$ et $y = -f_6 = 1$.