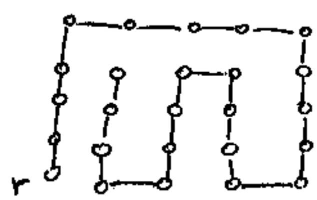
- Examen d'algos de graphes -

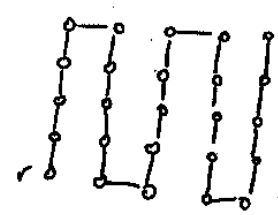
- Corrigé succinct -

Barême :
Exercice 1 sur 3.5 points : 1.5 - 0.5 - 1.5
Exercice 2 sur 6.5 points : 1.5 - 1.5 - 2 - 1.5
Exercice 3 sur 3 points : 1.5-1.5
Exercice 4 sur 8 points : 0.5 - 0.5 - 1 - 1 - 1 - 1.5 - 1.5 - 1

- Exercice 1 - Parcours en profondeur - 3.5pts -

1. et 2. Justes les dessins des parcours étaient demandés :





3. Pour le premier parcours, pour environ la moitié des sommets (on ne tient pas compte des sommets qui sont 'au bord'), on lit 'haut' et on va en haut : $\frac{n^2}{2}$ lectures, et pour environ l'autre moitié des sommets, on lit 'haut', 'droite', 'bas' et on va en bas : $3\frac{n^2}{2}$ lectures. En tout, on effectue donc environ $2n^2$ lectures.
   Et pour le second parcours, pour environ la moitié des sommets, on lit 'haut' et on va en haut : $\frac{n^2}{2}$ lectures, et pour environ l'autre moitié des sommets, on lit 'haut', 'bas' et on va en bas : $2\frac{n^2}{2}$ lectures. En tout, on effectue donc environ $\frac{3n^2}{2}$ lectures.

- Exercice 2 - Plus courts chemins - 6.5pts -

C'est quasiment que du cours...

1. Voir Cours ou TD. On obtient l'arborescence donné par le tableau *pere* et avec les distance d ci-dessous :

sommet	s	a	b	c	d
pere	s	s	c	s	c
d	0	6	10	7	9

   J'attendais une ou deux explications sur le calcul...
2. Rien que du cours...
3. On applique Bellman-Ford. L'ordre de traitement des arcs est *ac*, *ad*, *ba*, *cb*, *cd* *db*, *sa* puis *sc*. On obtient l'arborescence donné par le tableau *pere* et avec les distance d ci-dessous (après 4 passages dans la boucle principale) :

sommet	s	a	b	c	d
pere	s	b	c	s	a
d	0	2	4	7	-2

J'attendais quelques étapes de calcul.

4. Si le graphe contient un cycle de longueur totale négative, l'algorithme ne calculera pas les bonnes longueurs de plus courts chemins.
Pour détecter un cycle de longueur totale négative, on peut ajouter le tour de boucle supplémentaire donné en cours. Pour afficher de plus un tel cycle, on ajoute les lignes suivantes :

Algorithme : CYCLE NÉGATIF
début
 pour tous les $uv \in A$ **faire** **si** $d(v) > d(u) + l(uv)$ **alors**
  Afficher `'Il existe un cycle orienté de poids <0' qui est:`;
  Afficher `'u'`;
  $tmp \longleftarrow pere(u)$;
  tant que $tmp \neq u$ **faire**
   Afficher `'tmp'`;
   $tmp \longleftarrow pere(tmp)$;
fin

- Exercice 3 - (Re) plus courts chemins - 3pts -

1. On déroule l'algo. On obtient comme PCC de 1 à 3 le chemin 1,2,3 et comme PCC de 3 à 4 le chemin 3,4. Le graphe correspondant est un cycle à 4 élément étiqueté 1,2,3,4 en le parcourant (on a $PCC(i, j) = j$ ssi ij est une arête du graphe).
2. Le tableau PCC contient dans la case (i, j) le premier pas d'un plus court chemin de i à j. On peut remplir PCC à la main ou utiliser l'algorithme de Floyd-Warshall pour le calculer. On obtient :

$$PCC = \begin{pmatrix} \emptyset & 2 & 2 & 5 & 5 \\ 1 & \emptyset & 3 & 3 \text{ ou } 4 & 5 \\ 2 & 2 & \emptyset & 4 & 2 \text{ ou } 4 \\ 5 & 3 \text{ ou } 5 & 3 & \emptyset & 5 \\ 1 & 2 & 2 \text{ ou } 4 & 4 & \emptyset \end{pmatrix}$$

- Exercice 4 - Graphe sans triangle -8pts-

1. Un seul sommet, une seule arête, un chemin, un cycle pair convenait par exemple.
2. Un cycle à 5 sommet convenait par exemple.
3. Rapidement on testait tous les triplets de sommets et on regardait pour chacun si le triplet formait un triangle dans le graphe.
4. (a) On peut faire un peu mieux que précédemment :
   (b) Comme xy n'est contenu dans aucun triangle, x et y on des voisinnages disjoints. On peut écrire : $|N_{G\setminus\{x,y\}}(x)| + |N_{G\setminus\{x,y\}}(x)| \leq |V \setminus \{x, y\}| = n - 2$ et $d_G(x) + d_G(y) = 1 + |N_{G\setminus\{x,y\}}(x)| + 1 + |N_{G\setminus\{x,y\}}(y)| \leq 2 + n - 2 = n$.

Algorithme : Détection de triangles
début
pour tous les $uv \in E$ **faire** **pour tous les** $w \in v$ **faire** **si** $uw \in E$ *et* $uv \in E$ **alors**
Retourner `'Il existe un triangle dans le graphe: uvw'`;
fin

5. Pour un sommet u fixé, dans la somme $\sum_{uv \in E(G)}(d(u) + d(v))$, $d(u)$ apparaît autant de fois qu'un arête uv existe dans $E(G)$, c'es-à-dire exactment $d(u)$ fois. Comme cela est valide pour tout sommet u, on en déduit l'égalité attendue.
   On a donc $\sum_{u \in V(G)}(d(u))^2 = \sum_{uv \in E(G)}(d(u) + d(v)) \leq \sum_{uv \in E(G)} n = nm$.
6. On a ainsi $(\sum_{u \in V(G)} d(u))^2 = (\sum_{u \in V(G)} d(u).1)^2 \leq (\sum_{u \in V(G)}(d(u))^2).(\sum_{u \in V(G)} 1^2)$ par l'inégalité de Cauchy-Schwartz. D'où $(\sum_{u \in V(G)} d(u))^2 \leq nm.n$ par la question précédente. En uilisant l'égalité $\sum_{u \in V(G)} d(u) = 2m$, on obtient $4m^2 \leq mn^2$ et finalement $m \leq \frac{n^2}{4}$.
7. Un biparti complet donc chaque partie contient $\frac{n}{2}$ sommets a n sommets, est sans triangle et contient exactement $\frac{n^2}{4}$ arêtes.