- Contrôle continu - SUJET A -

- CORRIGÉ -
Novembre 2016

- Exercice 1 - ACPM *(3.5 points)* -

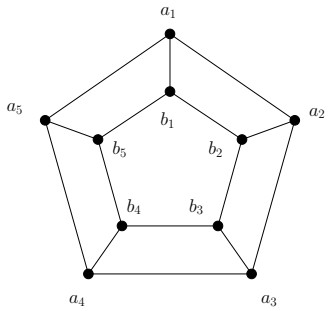
a. **(1.5 pts)** cf Cours...

b. **(1 pt)** On obtient l'arbre formé par les arêtes $\{ab, ad, ae, bc, dg, fg\}$.

c. **(1 pt)** On fait un parcours (largeur ou profondeur) sur le graphe formé des arêtes de poids minimum. On obtient un arbre couvrant de G ne contenant que des arêtes de poids min, c'est forcément un APCM. Former ce graphe se fait en temps linéaire et on obtient un algo en $O(m+n)$, meilleur que Krukal optimisé, en $O(m \log n)$.

- Exercice 2 - Parcours en profondeur *(3 points)* -

1. **(0.5 pt)**



2. **(1.5 pt)**
Parcours de type A (il y a plusieurs solutions possibles...) :
Pile AT : $(a_1, a_2, a_3, a_4, a_5, b_5, b_1, b_2, b_3, b_4)$

`sommet`	a_1	a_2	a_3	a_4	a_5	b_1	b_2	b_3	b_4	b_5
`pere`	a_1	a_1	a_2	a_3	a_4	b_5	b_1	b_2	b_3	a_5
`debut`	1	2	3	4	5	7	8	9	10	6
`fin`	20	19	18	17	16	14	13	12	11	15

Parcours de type B (il y a plusieurs solutions possibles...) :
Pile AT : $(a_1, b_1, b_2, b_3, b_4, b_5, a_5, a_4, a_3, a_2)$

`sommet`	a_1	a_2	a_3	a_4	a_5	b_1	b_2	b_3	b_4	b_5
`pere`	a_1	a_3	a_4	a_5	b_5	a_1	b_1	b_2	b_3	b_4
`debut`	1	10	9	8	7	2	3	4	5	6
`fin`	20	11	12	13	14	19	18	17	16	15

3. **(1 pt)** Par exemple : $AT = (a_1, a_2, b_2, b_3, b_4, b_5, a_5, a_4, a_3, b_1)$.

- Exercice 3 - Diamètre *(4 point)* -

1. **(1 pt)** Les sommets e et g sont à distance 3 l'un de l'autre. Les sommets a, b, c et d sont à distance au plus deux 2 de n'importe quel sommet. Pour finir e est à distance 2 de h et f. Ainsi, à symétrie près, e et g sont les sommets les plus éloignés, donc $\oslash(G_{CC}) = 3$.

2. **(1 pt)** File AT : (e, a, b, c, d, h, f, g)

sommet	a	b	c	d	e	f	g	h
pere	e	e	a	a	e	b	c	a
ordre	2	3	4	5	1	7	8	6
niveau	1	1	2	2	0	2	3	2

3. **(1.5 pt)**

Algorithme : diam(G)
début
  $diam \longleftarrow 0$;
  pour $r \in V(G)$ **faire**
    Effectuer un parcours en largeur issu de r, récupérer la fonction `niveau`;
    $tmp \longleftarrow$ valeur minimum du tableau `niveau`;
    si $tmp > diam$ **alors** $diam \longleftarrow tmp$;
  fin
  retourner $diam$;
fin

4. **(0.5 pt)** Le parcours en largeur demande un temps en $O(n+m)$, la recherche du minimum de `niveau` un temps en $O(n)$. On effectue n fois ces opérations donc une complexité de $O(n^2+nm)$ en tout. Comme G est connexe, on a $m \geq n-1$, donc la complexité de l'algo est, plus clairement, $O(mn)$.

- Exercice 4 - Algo mystère... *(3 points)* **-**

a. **(1 pt)** On choisit a, on marque (ie. on passe c à 1) les sommets a, b, c, d, e et h. Le premier sommet par ordre alphabétique non marqué est f. On sélectionne f, et fixe $c(f) = 1$. Le sommet g es choisit dans X, le dernier h à déjà $c(h) = 1$. Au final, on obtient $X = \{a, f, g\}$.

b. **(1 pt)** Les boucles ligne 3 et 4 s'exécutent n fois. La totalité des exécutions de la ligne 8 prend un temps $O(\sum_{x \in V} deg_G(x)) = O(m)$. En tout, on a un temps en $O(n+m)$. *(1/2 points pour une réponse en $O(n^2)$)*

c. **(0.5 pt)** A la fin de l'exécution de `ALGO`, X est un stable (maximal) de G.

d. **(0.5 pt)** `ALGO` aurait pu retourner par exemple l'ensemble $\{e, f, g, h\}$.

- Exercice 5 - Composantes connexes *(2.5 points)* **-**

Sinon G contient au moins 3 composantes connexes. On considère alors C la composante connexe de G ayant le plus petit nombre de sommet. Par hypothèse, on a donc $|C| \leq n/3$. Soit x un sommet de C. Comme les voisins de x sont tous dans C, x a au plus $|C|-1$ voisins. Donc on obtient $deg_G(x) \leq |C| - 1 \leq n/3 - 1$, ce qui contredit l'hypothèse de l'énoncé. *(pour avoir les 2.5 pts, il faut que la rédaction soit particulièrement claire...)*