Algorithmique avancée : Contrôle Continu n^{o}1

Vendredi 21 Octobre 2016. Durée 2h. Documents autorisés.

Le barème est donné à titre indicatif. Toutes les réponses doivent être soigneusement **justifiées**. La compréhension du sujet faisant partie de l'épreuve, **on ne répondra à aucune question**. Si vous rencontrez des ambiguïtés, vous expliquerez **sur votre copie** comment vous les interprétez.

I Complexité (4 points)

1. Un algorithme a une complexité temporelle qui peut s'exprimer sous la forme suivante :

$$T(n) = 27T(n/3) + 7n^3 + 4$$

Donnez une estimation asymptotique Θ de cette complexité.

2. On cherche à avoir une approximation de $\log(n!)$. Pour cela, on considère l'approximation de Stirling : $n! \sim \sqrt{2.\pi} n^{n+\frac{1}{2}} e^{-n}$ et on note :

$$f(n) = n^{n+\frac{1}{2}} e^{-n}$$

donnez une approximation asymptotique en Θ de la fonction $\log(f(n))$ puis conclure sur $\log(n!)$.

II Structures de données (8 points)

On considère le tableau $T1$ contenant les 19 éléments suivants :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
10	36	9	12	25	8	14	29	18	11	17	38	27	3	6	89	13	77	21

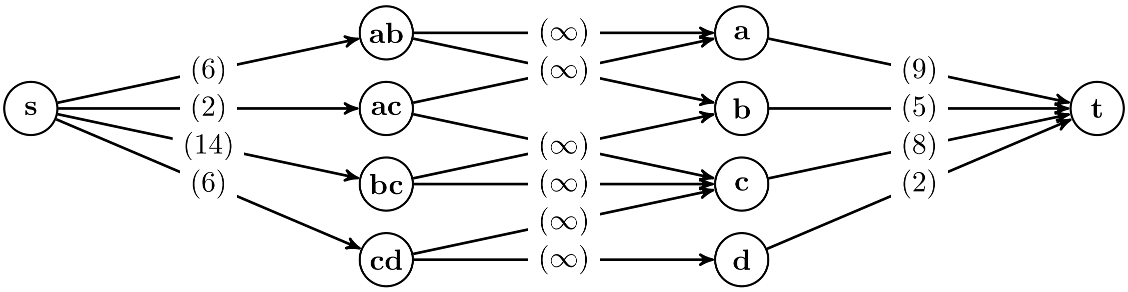
1. On désire créer un tas binomial à partir du tableau initial $T1$,
   a) Donnez le nombre d'arbres binomiaux avec leurs ordres pour un tas binomial contenant le nombre d'éléments du tableau $T1$.
   b) Créez la forêt $F1$ des arbres binomiaux décrits à la question précédente en remplissant les noeuds des arbres avec les éléments du tableau initial $T1$. Vous positionnerez les éléments au premier noeud vide du premier arbre binomial non plein en remplissant chaque arbre de gauche à droite et de haut en bas.
2. Cette forêt $F1$ n'est pas un tas binomial, quelles opérations faut-il faire pour la transformer en tas binomial ? Effectuez ces opérations et décrivez le tas binomial obtenu $F2$.
3. Combien de feuilles y a-t-il dans un arbre binomial d'ordre k ?
4. Quelle est la complexité en pire cas de toute la procédure de création de tas binomial que l'on vient d'effectuer en fonction de la taille initiale n du tableau ? (vous pouvez donner une expression en fonction du nombre de Fibonacci)
5. Dessinez le B-Arbre (B-Tree en anglais) de degré minimum $t = 4$ résultant de l'insertion successive des nombres du tableau $T1$. Vous dessinerez également les arbres intermédiaires avant et après chaque éclatement de noeuds.

III Le problème de sélection (8 points)

On dispose d'un ensemble O de 4 objets a, b, c et d, chacun a un coût, leurs coûts respectifs sont 9, 5, 8 et 2. Certaines combinaisons d'objets sont plus ou moins utiles, on considère donc que certains ensembles rapportent un gain, soit E l'ensemble des combinaisons qui rapportent. Ici, $\{a, b\}$ rapporte 6, $\{a, c\}$ rapporte 2, $\{b, c\}$ rapporte 14, $\{c, d\}$ rapporte 6.

L'objectif est de sélectionner un ensemble d'objets Y de façon à obtenir un bénéfice maximum (c'est-à-dire que la différence entre la somme des gains des combinaisons que forment les objets de Y et la somme des coûts des objets de Y, soit maximale).

On représente ce problème par le réseau suivant :



1. a) Donnez un exemple non trivial de coupe de capacité finie (c'est-à-dire différent de $(\{s\}, X \setminus \{s\})$ et de $(X \setminus \{t\}, t)$ avec sa capacité.
   b) Les coupes de capacité finies correspondent à la sélection de certains objets, donnez pour votre coupe les objets sélectionnés. Calculez le gain et le coût associé à cette sélection.
2. Calculez une coupe de capacité minimum sur ce réseau. Vous détaillerez les étapes du calcul et la méthode utilisée. Listez ses arcs.
3. Soit A un ensemble de sommets tel que $(A, X \setminus A)$ est une coupe de capacité finie.
   a) Montrez que $\forall u \in \omega^+(A)$, on a $u = (s, e)$ ou bien $u = (o, t)$ avec e un sommet représentant une combinaison d'objets ($e \in E$) et o un objet ($o \in O$).
   b) Soit une coupe de capacité finie $C = (A, X \setminus A)$, et soit Y l'ensemble de tous les objets apparaissant dans les arcs d'extrémité t dans cette coupe. Montrez que
   $$capa(C) = \sum_{o \in Y} cout(o) + \sum_{e \in E_{\overline{Y}}} gain(e)$$
   où $E_{\overline{Y}} = \{e \in E \text{ et } e \not\subseteq Y\}$ est l'ensemble des sommets représentant des combinaisons dont les objets ne sont pas tous dans Y.
   c) En déduire que
   $$capa(C) = \sum_{o \in Y} cout(o) + \sum_{e \in E} gain(e) - \sum_{e \in E, e \subseteq Y} gain(e)$$
   et que l'on peut réécrire cette égalité en
   $$capa(C) = \sum_{e \in E} gain(e) - gainNet(Y)$$
   vous direz ce que représente $gainNet(Y)$.
   d) En déduire que maximiser ce que rapporte une sélection d'objet Y revient à sélectionner une coupe de capacité minimum. Quels objets faut-il sélectionner ? Pour quel bénéfice ?