

Test de niveau en algorithmique

Christian Rinderknecht

Mardi 4 mars 2003

1 Binôme de Pascal

Soit la fonction `binome` spécifiée ainsi :

```
binome (n,p) {  
  if p = 0 or p = n then return 1  
  else return binome (n-1,p) + binome (n-1,p-1);  
}
```

1. Déterminez le nombre d'appels de la fonction `binome` effectués lors du calcul de `binome (n,p)` en fonction de `n` et `p`.
2. Peut-on augmenter la vitesse de calcul ?

2 Complexité et notations de Landau

1. Qu'est-ce que la complexité spatiale et temporelle d'un algorithme ?
2. Définissez les notations O , Ω et Θ et leur convention d'usage.

3 Tris

1. Qu'est-ce qu'une relation de comparaison ? Un ordre total ?
2. Qu'est-ce qu'une liste triée ?
3. Qu'est-ce qu'un algorithme de tri par comparaisons ?
4. Qu'est-ce qu'un algorithme de tri interne (ou *sur place*) ?
5. Qu'est qu'un algorithme de tri externe ?
6. Qu'est qu'un algorithme de tri stable ?
7. Présentez deux algorithmes de tri. Montrez leur terminaison et leur correction. Donnez leur complexité temporelle et spatiale dans le pire des cas. Discutez leur stabilité.
8. **Polynômes creux.** On représente les polynômes à une variable à coefficients entiers par des listes chaînées de monômes, un monôme ax^n étant représenté par le couple d'entiers (a, e) . Les monômes d'un polynôme sont classés par degré croissant et chaque monôme a un coefficient non nul. Programmez l'addition de deux polynômes pour cette représentation.

4 Arbres

4.1 Arbres binaires

1. Définissez les arbres binaires.
2. Qu'est-ce que le parcours en profondeur et en largeur ?
3. Parcours préfixe, postfixe et infixé ?
4. **Dénombrements sur les arbres binaires.** Établir :
 - (a) Un arbre binaire à n nœuds possède $n + 1$ branches vides.
 - (b) Dans un arbre binaire à n nœuds, le nombre de nœuds sans fils est inférieur ou égal à $(n + 1)/2$. Il y a égalité si et seulement si tous les nœuds ont zéro ou deux fils.
 - (c) La hauteur d'un arbre binaire non vide à n nœuds est comprise entre $\lfloor \log_2 n \rfloor$ et $n - 1$.
5. Qu'est-ce qu'un arbre binaire équilibré en hauteur ?

4.2 Arbres binaires de recherche

1. Qu'est-ce qu'un arbre binaire de recherche (ABR) ?
2. Écrire une fonction qui teste si un arbre binaire est bien un ABR.
3. Spécifiez l'ajout dans un ABR aux feuilles et à la racine. Comparez leur complexité temporelle et spatiale dans le pire des cas. Quel rapport y a-t-il entre un ABR créé par ajout aux feuilles et un par ajout à la racine, à partir des mêmes éléments ?
4. Programmez la suppression et la recherche dans un ABR. Quels sont leur coûts asymptotiques ?
5. Écrire une fonction réalisant la fusion de deux ABR. Analysez sa complexité temporelle.

5 Tables de hachage

1. Qu'est-ce qu'une table de hachage ? Dans quelles circonstances sont-elles une structure de donnée utile ?
2. Pourquoi recommande-t-on souvent de choisir un nombre premier pour la taille d'une table de hachage ?
3. Soient un ensemble E de n éléments et une fonction $h : E \rightarrow [1..m]$ uniforme (c-à-d. $\forall e \in E. \forall i \in [1..m]. \mathcal{P}\{h(e) = i\} = 1/m$). Montrez que la probabilité P que h soit injective vaut $m!/(m-n)!m^n$. En particulier, si $m = 356$ et $n = 23$, alors $P < 1/2$. Parieriez-vous que deux élèves dans cette classe soient nées le même jour du même mois ? Qu'en conclure à propos des tables de hachage ?
4. Expliquez les méthodes de résolution des collisions par chaînage interne et externe (*hachage indirect*), et par calcul (*hachage direct*).

6 Indexation

On désire écrire un programme qui, à la lecture d'un programme, repère pour chaque identificateur les lignes où il apparaît. Un identificateur commence par une lettre et n'est composé que de lettres et de chiffres. Certains mots, dits réservés, ne peuvent pas être utilisés comme identificateurs.

Après la lecture, on demande d'imprimer le programme, puis la liste (triée par ordre alphabétique) des identificateurs suivis chacun de la liste des numéros de ligne où il apparaît, ces numéros de ligne étant rangés en ordre croissant.

1. Faites une analyse comparée des différentes structures de données possibles ;
2. programmez une méthode utilisant des arbres binaires de recherche, et une méthode utilisant un tableau de hachage ; analysez les résultats.

7 Graphes

1. Qu'est-ce qu'un graphe non-orienté ? Orienté ?
2. Qu'est-ce qu'un graphe connexe ? Fortement connexe ?
3. Qu'est-ce qu'un parcours en profondeur d'abord ? En largeur d'abord ?
4. Donnez un algorithme de calcul des composantes fortement connexes, et sa complexité temporelle et spatiale dans le pire des cas.
5. Citez des cas où les graphes sont la structure de donnée idoine.

8 Automates et langages réguliers

1. Qu'est-ce qu'un automate fini ?
2. Qu'est-ce qu'un automate fini déterministe et non-déterministe ? Quels sont les liens entre ces deux types d'automates ?
3. Qu'est-ce qu'un automate minimal ? Y a-t-il unicité ? Donnez un algorithme de minimisation et discutez sa complexité.
4. Qu'est-ce qu'un langage régulier ? Une expression régulière ?
5. Quels liens y a-t-il entre les langages réguliers et les automates finis ?
6. Citez des applications intéressantes des automates finis et des expressions régulières.