Agrégation de Mathématiques option Informatique TP4

Loïg Jezequel

Mardi 7 Février 2012

1 Pour s'échauffer

Question 1. Écrivez une fonction triant une liste d'entiers, du plus petit au plus grand. Vous utiliserez l'algorithme de votre choix. Votre fonction se comporte-t-elle bien aux bornes ? (tri de la liste vide, d'une liste à un seul élément, d'une liste contenant plusieurs fois le même élément)

Question 2. Comment faire pour rendre votre fonction plus polyvalente ? (par exemple pouvoir trier les entiers selon un autre ordre, ou pouvoir trier autre chose que des entiers)

2 Le vif du sujet

Il s'agit de résoudre chaque exercice en 30 à 40 minutes.

Exercice 1 : empaquetage

On considère n objets de tailles $t_1, \ldots, t_n \in \mathbb{N}$, que l'on souhaite ranger dans des boîtes de capacité $C \in \mathbb{N}$, en s'assurant que la somme des tailles des objets rangés dans une boîte ne dépasse jamais cette capacité C. Le problème de l'empaquetage consiste à ranger les n objets en utilisant le moins de boîtes possible. Il s'agit d'un problème NP-difficile. Cependant il existe des algorithmes approchés, parmis lesquels :

NextFit On prend les objets dans l'ordre où ils sont donnés (de 1 à n) et on en met le plus possible dans la première boîte, puis dans la seconde, et ainsi de suite, sans jamais revenir sur les boîtes déja traitées.

FirstFit On prend les objets dans l'ordre où ils sont donnés et on les range dans la première boîte où il reste suffisamment de place.

BestFit On prend les objets dans l'ordre où ils sont donnés et on les range dans la boîte la plus pleine qui peut les contenir (en cas dégalité entre deux boîtes on prend la première des deux).

Question 3. Écrivez une fonction réalisant chacun de ces algorithmes.

Question 4. Peut-on améliorer ces algorithmes en triant les objets avant de les ranger ?

Exercice 2 : marches aléatoires

Une marche aléatoire y de longueur n est une suite de $n+1$ entiers $y(0), \ldots, y(n)$ telle que : $y(0) = 0$ et $|y(i+1) - y(i)| = 1$, $\forall 0 \leq i < n$.

Question 5. Écrivez une fonction qui génère une marche aléatoire.

On appelle *record* d'une marche aléatoire y tout indice $r > 0$, tel que $y(i) < y(r)$ pour tout $i < r$.

Question 6. Écrivez une fonction qui compte le nombre de records d'une marche aléatoire.

Question 7. Écrivez une fonction qui calcule la distance maximale entre deux records consécutifs d'une marche aléatoire.

On appelle *sous-suite strictement croisante* de longueur ℓ d'une marche aléatoire y toute suite d'indices $0 \leq i_1 < i_2 < \cdots < i_\ell \leq n$ telle que $y(i_1) < \cdots < y(i_\ell)$.

Question 8. Écrivez une fonction qui calcule la longueur des plus longues sous-suites strictement croissantes d'une marche aléatoire.

Exercice 3 : voyageur de commerce

On considère un ensemble de villes, représentées par des points du plan euclidien. La distance entre deux villes est la distance euclidienne. La longueur d'un parcours $v_0, v_1, \ldots, v_n$ est la somme des distances entre ses villes : $d(v_0, v_1) + d(v_1, v_2) + \cdots + d(v_{n-1}, v_n)$. Le problème est le suivant : un voyageur de commerce doit partir d'une ville de départ (v_0), visiter chacune des autres villes (une fois et une seule), et revenir ensuite à la ville v_0. Quelle est la permutation des villes qui engendre le parcours de longueur minimale ? Ce problème est connu pour être de complexité non-polynomiale. On s'intéresse ici à tester quelques heuristiques permettant de trouver de «bonnes» solutions en temps polynomial.

Méthode par insertion On part du chemin entre les villes v_0 et v_1. On insère la ville v_2 dans ce chemin, de manière à minimiser la longueur. On continue ainsi pour les villes suivantes. Si plusieurs positions sont possibles pour une ville on en choisit une arbitrairement.

Méthode itérative On part d'un parcours quelconque $v_{i_0}, v_{i_1}, \dots, v_{i_n}$ tel que $v_{i_0} = v_0$, passant une et une seule fois par chaque ville, et tel que $v_{i_n} = v_0$. On considère deux couples de villes voisines dans ce parcours : $(v_{i_k}, v_{i_{k+1}})$ et $(v_{i_\ell}, v_{i_{\ell+1}})$, tels que $0 \leq k < k+1 < \ell < \ell+1 \leq n$. On dit que l'on échange les deux segments de rangs k et ℓ si l'on transforme le parcours en allant de v_{i_k} à v_{i_ℓ} et de $v_{i_{k+1}}$ à $v_{i_{\ell+1}}$, et en renversant le sens du parcours initial entre v_{i_ℓ} et $v_{i_{k+1}}$. Un échange qui fait décroître la longueur du parcours est dit intéressant.

L'idée est de partir d'un parcours et d'appliquer successivement des échanges intéressants, jusqu'à ne plus en trouver.

Question 9. Écrivez une fonction réalisant chacun de ces algorithmes.

3 Encore du temps ?

Exercice 4 : recherche d'un zéro de fonction

Le but est, étant donnée une fonction f continue ayant des valeurs positives et des valeurs négatives, d'en trouver un zéro. On s'intéresse à deux méthodes :

Méthode par dichotomie On part d'un intervalle $[a_1, b_1]$ dans lequel on sait qu'il existe un zéro (pour faire simple on prendra $f(a_1)$ et $f(b_1)$ de signes différents). A chaque étape on coupe $[a_k, b_k]$ en deux parties égale et on en garde une, $[a_{k+1}, b_{k+1}]$, qui contient un zéro. On itère ce processus jusqu'à être aussi proche que souhaité d'un zéro de la fonction.

Méthode de la fausse position La méthode est similaire à la précédente. La seule différence est la manière de diviser l'intervalle : on le coupe au point correspondant à l'intersection entre l'axe des abscisses et la droite passant par $f(a_k)$ et $f(b_k)$.

Question 10. Écrivez pour chaque méthode une fonction qui, étant donnée une fonction f et un intervalle de départ $[a_1, b_1]$, retourne une approximation à ϵ près (ϵ sera aussi un paramètre de vos fonctions) d'un zéro de f.