Lycée La Prat's
Classe de PTSI

Jeudi 15 mai 2014

Épreuve d'Informatique 4

Durée 1 h

L'utilisation des calculatrices n'est pas autorisée pour cette épreuve.
Nous attacherons la plus grande importance à la lisibilité du code produit par les candidats; aussi, nous encourageons les candidats à introduire des procédures ou fonctions intermédiaires lorsque cela en simplifie l 'écriture.

Une copie par exercice. Ne passez pas trop de temps sur la question 3 de l'exercice 1, vous pourrez y revenir après au besoin. L'énoncé est long car la présentation de l'exercice 2 est très détaillée.

Exercice 1

Soit L une liste d'entiers relatifs, de taille N.

1. Écrire une fonction qui prend en argument une liste `L` d'entiers et retourne un liste `resultat` des sommes partielle $\sum_{i=0}^{k} \ell_i$: la k-ème case de `resultat` contiendra la somme `L[0]+...+L[k]`. On essayera (sans y passer trop de temps non plus) d'être le plus efficace (en terme de complexité) possible.
2. Écrire une fonction qui prend en argument une liste `L` d'entiers et retourne la sous-somme $\sum_{i=0}^{k} \ell_i$ maximale.
3. Écrire une fonction qui prend en argument une liste `L` d'entiers et retourne la sous-somme $\sum_{i=k_1}^{k_2} \ell_i$ maximale (on commence à k_1 quelconque désormais).

Exercice 2 (Poteaux télégraphiques)

Cette exercice a pour objectif de choisir où placer des poteaux télégraphiques pour relier le point le plus à gauche d'un paysage unidimensionnel au point le plus à droite en fonction de critères de coût. Nous ferons les simplifications suivantes : les fils sont sans poids et tendus ; ils relient donc en ligne droite les sommets de deux poteaux consécutifs. Les normes de sécurité imposent que les fils soient en tout point à une distance d'au moins Δ (mesurée verticalement) au-dessus du sol. Les poteaux sont tous de longueur identique $\ell \geqslant \Delta$. Voici par exemple une proposition valide de placement de poteaux pour le paysage ci-dessous avec $\Delta = 0.5$ et $\ell = 2.0$.

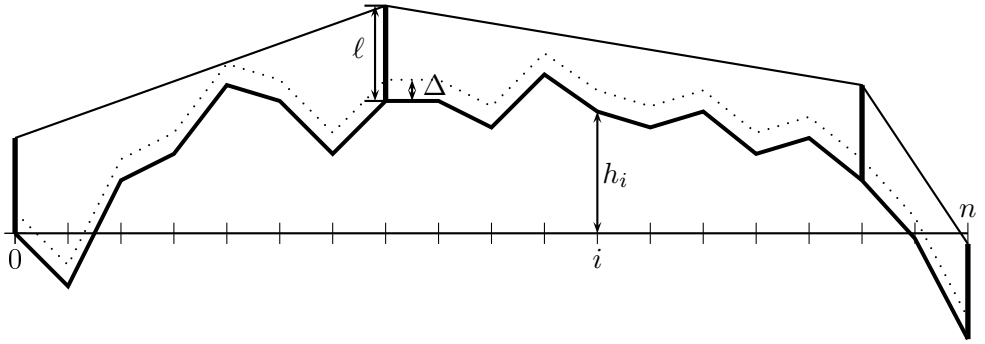

FIGURE 1 – Un exemple de poteaux où le fil est en tout point au moins Δ au-dessus du sol.

Tous les tableaux définis dans ce problème seront considérés comme des variables globales, accessibles et modifiables directement par toutes les procédures et fonctions sans avoir à les passer comme arguments.

Partie I. Planter le paysage

Nous définissons le paysage à partir d'une suite de relevés de dénivelés stockés dans un tableau `deniveles` de nombres flottants (c'est-à-dire des nombres à virgule) de taille $n + 1$. Le tableau `deniveles` est supposé être une variable globale : il n'est pas nécessaire de le passer en paramètre, il accessible depuis toutes les fonctions.
Le paysage est alors décrit par une ligne brisée de $n + 1$ points, dont le i-ème point P_i est de coordonnées (i, h_i) où :

$$h_0 = 0.0 \text{ et } h_i = h_{i-1} + \texttt{deniveles}[i] \text{ pour } i \in \{1, ..., n\}.$$

Ainsi, la hauteur du i-ème point est la hauteur du $(i-1)$-ème point plus la dénivelée stoquée en `deniveles[i]`.

Question 1 Écrire une fonction `calculHauteurs`(n) qui retourne un tableau de taille $n + 1$ contenant les hauteurs h_i des points. On suppose désormais que ce résultat est stocké dans une variable globale `hauteurs`.

Question 2 Écrire une fonction `calculFenetre`(n) qui calcule les hauteurs minimale et maximale que peut atteindre un point du paysage. Elle calculera également les indices de points les plus à gauche de hauteur minimale et maximale.
On supposera que les résultats sont stockés dans des variables globales `hMin` et `hMax` (pour les hauteurs) et `iMin` et `iMax` (pour les indices).
On appelle *distance* au sol de i à j, la longueur de la ligne brisée allant du point P_i au point P_j.

Question 3 Écrire une fonction `distanceAuSol`(i, j) qui calcule et renvoie la distance au sol de P_i à P_j. On pourra commencer par calculer la distance au sol entre P_i et P_{i+1}, à l'aide du théorème de Pythagore. On suppose que h_i est donnée en mètres, de même que les abscisses i (horizontalement, chaque point est distant d'un mètre du suivant).
On appelle un *pic* un point P_i d'indice $1 \leqslant i < n$ tel que $h_i > \max(h_{i-1}, h_{i+1})$. On appelle *point remarquable*, les pics et les points des bords gauche (point P_0) et droit (point P_n). On appelle *bassin* toute partie du paysage allant d'un point remarquable au suivant.

Question 4 Écrire une fonction `pointRemarquable`(i) qui détermine si le point P_i est remarquable, et renvoie un booléen.

Question 5 Écrire une fonction `longueurDuPlusLongBassin`(n) qui calcule et renvoie la longueur au sol maximale d'un bassin dans le paysage.

Partie II. Planter les poteaux

On souhaite relier le point le plus à gauche au point le plus à droite par un fil télégraphique. Pour cela, nous devons choisir à quels points parmi les $(P_i)_{0\leqslant i\leqslant n}$ planter les poteaux télégraphiques intermédiaires. Rappelons que le fil est supposé sans poids et tendu et qu'il relie donc les sommets de chacun des poteaux en ligne droite. Les poteaux sont plantés verticalement et ont tous une longueur identique $\ell \geqslant \Delta$ (ℓ et Δ sont des nombres flottants).
La législation impose que le fil doit rester à une distance supérieure ou égale à Δ (mesurée verticalement) au-dessus du sol. Pour tester si un fil tiré entre un poteau placé au point P_i et un poteau placé au point P_j (avec $j > i$ ou $j < i$) respecte la législation, notons

$$\alpha_{i,k} = \frac{(h_k + \Delta) - (h_i + \ell)}{k - i} \text{ pour } k \neq i, \text{ et } \beta_{i,j} = \frac{(h_j + \ell) - (h_i + \ell)}{j - i},$$

les pentes des fils tirés depuis le poteau en P_i jusqu'à une hauteur Δ au-dessus du point P_k d'une part et jusqu'à une hauteur ℓ au-dessus du point P_j d'autre part (voir la Figure 2).
On admet que le fil tiré d'un poteau en P_i à un poteau en P_j respecte la législation si et seulement si :

$$\beta_{i,j} \geqslant \max_{i<k<j} \alpha_{i,k}, \text{ lorsque } j > i; \text{ et } \beta_{i,j} \leqslant \min_{j<k<i} \alpha_{i,k}, \text{ lorsque } j < i.$$

c'est-à-dire,
- lorsque $j > i$, $\beta_{i,j}$ est plus grand que tous les $\alpha_{i,k}$ lorsque k parcourt $[\![i + 1, j - 1]\!]$
- ou bien, lorsque $j < i$, $\beta_{i,j}$ est plus petit que tous les $\alpha_{i,k}$ lorsque k parcourt $[\![i + 1, j - 1]\!]$.

Question 5 En utilisant cette méthode, écrire une fonction `estDeltaAuDessusDuSol`$(i, j, \ell, delta)$ qui renvoie `True` si un fil tiré entre les sommets d'un poteau placé au point P_i et d'un poteau placé au point P_j respecte la législation, et renvoie `False` dans le cas contraire.

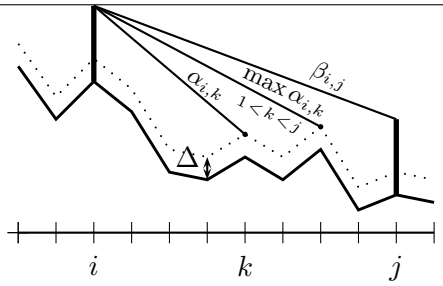

FIGURE 2 – Condition sur les pentes ($\beta_{i,j}$, les $\alpha_{i,k}$) pour pouvoir tirer un fil.

Considérons une première stratégie, dite *algorithme glouton* en avant. Le premier poteau est planté en P_0. Pour calculer l'emplacement du prochain poteau, on part du dernier poteau planté et on avance (à droite) avec le fil tendu tant que la législation est respectée (et que P_n n'est pas atteint). Un nouveau poteau est alors planté, et on recommence jusqu'à ce que P_n soit atteint. La figure 3 illustre la solution produite par cet algorithme.

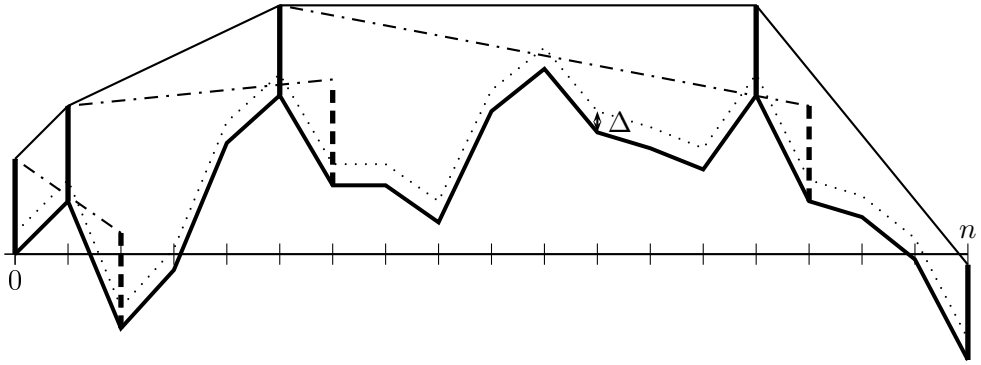

FIGURE 3 – Solution produite par l'algorithme glouton en avant. Les poteaux et les fils en pointillés sont ceux violant la législation.

Question 6 Écrire une fonction `placementGloutonEnAvant`$(n, \ell, delta)$ qui calcule la disposition des poteaux selon l'algorithme ci-dessus. La solution sera stockée dans un tableau `poteaux` de la façon suivante :

- la case `poteaux`[0] contiendra le nombre de poteaux utilisés ;
- la case `poteaux`$[t]$, pour $t \geqslant 1$, contiendra l'indice i indiquant que le t-ème poteau, s'il existe, est planté en P_i

La figure 3 correspond au tableau `poteaux` suivant : `[5,1,5,14,18]`.
Le nombre de poteaux est stocké dans `poteaux[0]`, qui vaut donc ici 5. Le premier poteau est toujours en 0 : il n'est pas noté.

Question 7 Donner une majoration de la complexité du temps de calcul de votre algorithme en fonction de n. Justifier qu'on peut se contenter du calcul de $O(n)$ pentes pour implémenter l'algorithme glouton en avant. Expliquer succinctement comment modifier votre fonction (si nécessaire) pour obtenir une complexité en $O(n)$. *Aucune implémentation n'est demandée dans cette question.*