



ANNÉE UNIVERSITAIRE 2024-2025
SESSION 1 DE PRINTEMPS

Parcours/Étape: L3 Info./MI/CMI Code UE: 4TIN602U

Épreuve: Techniques Algorithmiques et Programmation

Date: 22/04/2024

Heure: 9h00

Durée: 1h30

Documents: une seule feuille A4 recto-verso autorisée.

Épreuve de M. Cyril GAVOILLE

université
de BORDEAUX

Collège Sciences
et Technologie

RÉPONDRE DIRECTEMENT SUR LE SUJET
QUI EST À RENDRE DANS LA FEUILLE DOUBLE D'EXAMEN

Question de cours

Dans l'algorithme ApproxMST, permettant d'approximer la solution optimale du VOYAGEUR DE COMMERCE (V, d) , nous avons été amenés à considérer le graphe complet à n sommets où les sommets correspondent aux n points de V , et les arêtes à toutes les distances entre les points de V .

Question 1. Dessiner le graphe complet à $n = 6$ sommets et donnez son nombre d'arêtes.

Question 2. Donnez la formule générale donnant le nombre d'arêtes du graphe complet à n sommets. Justifiez.

Question 3. Ordonnez de manière croissante les 9 complexités asymptotiques suivantes :

$$2^n, \ln(n) \ln \ln(n), n, e^n, e^{n/\ln(n)}, n^{(\ln(n))^2}, n \ln(n), n^2, n\sqrt{n}$$

Vous pourrez écrire $f(n) < g(n)$ pour dire que la complexité $f(n)$ est asymptotiquement plus petite que la complexité $g(n)$.

Question 4. Deux informaticiens ont chacun conçu un nouvel algorithme de tri. On s'intéresse à leur complexité asymptotique dans le pire des cas en nombre de comparaisons. Le premier affirme que son algorithme a une complexité bien précise de $f(n) = \lceil 2n \ln(n) \rceil$, alors que le deuxième annonce que son algorithme a une complexité $g(n) = O(n \ln(n))$. En terme de complexité, y a-t'il un algorithme meilleur que l'autre ? Si oui, lequel ? Discutez.

Mine d'or

Un mineur souhaite extraire le meilleur filon d'une mine d'or. Une étude précise du terrain a permis de mesurer la quantité d'or disponible en kg dans chaque bloc de la mine, les blocs étant répartis en couches linéaires niveau par niveau. Le résultat est donné sous la forme d'une matrice M rectangulaire $H \times L$, où $M(i, j)$ indique la quantité d'or disponible à la profondeur $i \in \{0, \dots, H - 1\}$ et à la position $j \in \{0, \dots, L - 1\}$. On supposera $H, L \geq 1$.

Pour simplifier, on identifiera la mine avec sa matrice de mesures. Voici un exemple M_0 de mine 4×3 .

$$M_0 = \begin{array}{ccc|l} & j = 0 & j = 1 & j = 2 & \\ \hline & 1 & 2 & \mathbf{5} & i = 0 \\ & 7 & \mathbf{5} & 0 & i = 1 \\ & 6 & \mathbf{7} & 4 & i = 2 \\ & \mathbf{2} & 0 & 1 & i = 3 \end{array}$$

Le processus d'extraction d'un filon d'or dans M est le suivant. Le niveau $i = 0$ correspond à la surface de la mine et est immédiatement accessible au mineur. Le mineur est donc libre de choisir la position initiale comme il le souhaite. Une fois cette position j choisie, il creuse le bloc pour récupérer la quantité $M(i, j)$ d'or qui s'y trouve. De là, il peut accéder à l'un des blocs de niveau $i + 1$ qui est soit juste en dessous, juste avant ou juste après la position j . Dit autrement, une fois le bloc (i, j) extrait, il peut extraire l'un des blocs $(i + 1, j - 1)$, $(i + 1, j)$ ou $(i + 1, j + 1)$. Il continue ainsi de suite jusqu'à atteindre le fond de la mine. Un filon peut être vu comme un chemin de blocs reliant la première à la dernière ligne, qui ne fait que descendre avec des mouvements latéraux de -1 , 0 ou $+1$.

Le but du mineur est d'extraire le meilleur filon de M , c'est-à-dire celui contenant la quantité maximum d'or. Pour M_0 , le meilleur filon contient $5 + 5 + 7 + 2 = 19$ kg d'or et est composé des blocs $(0, 2) \rightarrow (1, 1) \rightarrow (2, 1) \rightarrow (3, 0)$.

Pour un bloc (i, j) d'une mine M donnée, on définit $f(i, j)$ comme la quantité d'or du meilleur filon à partir du bloc (i, j) . On a vu pour M_0 que $f(0, 2) = 19$. On a aussi $f(1, 1) = 5 + 7 + 2 = 14$ et $f(2, 2) = 4 + 1 = 5$.

Question 5. Pour la mine M_0 , déterminer $f(1, 0)$, $f(0, 0)$, $f(0, 1)$, en précisant les calculs.

Il est clair que le meilleur filon consiste à choisir une position initiale j telle que $f(0, j)$ est maximum. Autrement dit, la quantité d'or $Q(M)$ du meilleur filon vaut précisément

$$Q(M) = \max_{j \in \{0, \dots, L-1\}} f(0, j)$$

Grâce à la question 5, vous pouvez vérifier en effet que $Q(M_0) = \max \{f(0, 0), f(0, 1), f(0, 2)\} = 19$.

Question 6. Pour une mine M quelconque, déterminez une formule de récurrence permettant de calculer $f(i, j)$. [Pour simplifier, posez $f(i, j) = 0$ si $i \notin \{0, \dots, H-1\}$ ou $j \notin \{0, \dots, L-1\}$.]

Question 7. Déterminez le nombre de nœuds de l'arbre des appels de $f(0, 0)$ pour la mine M_0 et dessinez l'arbre correspondant. Pour simplifier, vous pourrez représenter l'appels (i, j) par " ij ". [Aide : Représentez uniquement les appels valides et commencez par les (sous-)arbres de $f(1, 0)$ et $f(1, 1)$.]

Question 8. De manière générale, en fonction de H et L , donnez un majorant pour le nombre de nœuds de l'arbre des appels de $f(0, j)$ pour une position j quelconque. Justifiez.

Question 9. En déduire la complexité du calcul de la quantité d'or du meilleur filon $Q(M)$ en utilisant le calcul récursif de $f(i, j)$.

Il est facile de voir qu'avec cette méthode récursive pour $Q(M)$ il existe des calculs inutiles. Par exemple, le calcul de $f(0, 0)$ utilise $f(1, 0)$ et $f(1, 1)$, mais le calcul de $f(0, 1)$ utilise aussi ces deux valeurs. On se propose donc d'employer la programmation dynamique. Pour cela, on va utiliser un tableau `double F[H][L]`; où $F[i][j] = f(i, j)$ tout simplement.

Question 10. Donnez le tableau `F` rempli pour la mine M_0 . [Aide : La dernière ligne est facile à construire et vous devriez avoir $F[0][2] = 19$.]

Question 11. En supposant que `H` et `L` sont des variables globales, écrivez la partie manquante du code ci-dessous censé calculer par programmation dynamique $Q(M)$ pour une mine M quelconque. [Aide : Faites attention aux bords de la mine et au cas $L = 1$.]

```
int const H,L;
double Q(double M[H][L]){
    double F[H][L];
    ...; // partie à compléter dans le cadre ci-dessous
    int q=0;
    for(int j=0; j<L; j++) q = fmax(q, F[0][j]);
    return q;
}
```

Question 12. *Quelle est la complexité de $\mathcal{Q}(\mathbf{M})$ en fonction des dimensions de \mathbf{M} ? Justifiez.*

Seaux d'eau

On dispose d'une batterie de n seaux sans graduation. La capacité est variable suivant le seau mais est toujours un nombre entier de litres. Par exemple $(4, 3, 1)$ représente une batterie de $n = 3$ seaux de capacités 4, 3 et 1 litres respectivement.

Initialement, le premier seau de la batterie est rempli d'eau, tous les autres étant vides. Une configuration est une suite qui indique le nombre de litres d'eau contenu dans chacun des seaux. Donc, pour la batterie $(4, 3, 1)$, la configuration initiale est $(4, 0, 0)$.

L'objectif est de savoir s'il est possible, à l'aide d'opérations de transvasement, d'obtenir une configuration donnée. Par exemple, à partir de la configuration $S = (4, 0, 0)$ on peut obtenir la configuration $T = (2, 2, 0)$ en versant le 1er seau dans le 2e, ce qui produit la configuration $(1, 3, 0)$, puis en versant le 2e seau dans le 3e, ce qui produit $(1, 2, 1)$, et enfin en versant le 3e seau vers le 1er. Cela produit ainsi un chemin avec trois versements : $(4, 0, 0) \rightarrow (1, 3, 0) \rightarrow (1, 2, 1) \rightarrow (2, 2, 0)$. Bien entendu, les versements doivent s'effectuer complètement, sans perte d'eau et en respectant la capacité des seaux, c'est-à-dire que le transvasement s'arrête dès que le seau qu'on remplit est plein ou que le seau qu'on vide est vide.

En **C** on aurait pu écrire :

```
#define n 3
typedef int batterie[n];
typedef int config[n];
batterie B={4,3,1};
config S={4,0,0};
config T={2,2,0};
```

Étant donnée une batterie $B = (b_0, \dots, b_{n-1})$ de n seaux, on définit le graphe G_B comme le graphe des configurations atteignables depuis la configuration initiale $S = (b_0, 0, \dots, 0)$, deux configurations $X = (x_0, \dots, x_{n-1})$ et $Y = (y_0, \dots, y_{n-1})$ de G_B étant connectées par un arc $X \rightarrow Y$ si $Y \neq X$ et s'il existe un transvasement du seau i vers le seau j pour un certain couple $(i, j) \in \{0, \dots, n-1\}^2$.

Question 13. *Dessinez le graphe G_B pour $B = (4, 3, 1)$. Le graphe est-il symétrique ? (c'est-à-dire, pour tout X, Y , si $X \rightarrow Y$ alors $Y \rightarrow X$). [Aide : Le graphe comporte huit configurations.]*

Pour trouver un chemin d'une configuration S à T dans G_B avec le moins de transvasements possibles, on se propose d'utiliser l'algorithme A^* vu en cours. Pour cela, on a besoin de générer tous les voisins d'un sommet donné.

Question 14. Écrivez la fonction `int successors(config X, config L[])` qui étant donnée une configuration X de G_B renvoie dans L la liste des successeurs de X , c'est-à-dire toutes les configurations Y de G_B telles que $X \rightarrow Y$. En valeur de retour, la fonction doit renvoyer le nombre de successeurs de X . [Aide : Attention ! S'il y a un transvasement, on doit obtenir une configuration différente de X .]

Question 15. Quelle est la complexité de votre fonction ? Justifiez.

En combinaison avec A^* , on propose d'utiliser l'heuristique h suivante : $h(X, T)$ est le nombre de seaux de la configuration X incorrectement remplis dans T . Par exemple, pour la batterie $B = (4, 3, 1)$, $X = (4, 0, 0)$ et $T = (2, 2, 0)$, nous avons $h(X, T) = 2$.

Question 16. Démontrez que l'heuristique h n'est pas monotone.

Question 17. Proposez une variante de l'heuristique précédente afin que A^* trouve un plus court chemin de S à T , s'il existe.

FIN.