

AHOSAKADI KWALAKANA (2GC)

MULAMBA LUBANZA (2GEI)

TSHISENSE NGANDU (2GC)

DEVOIR N°02 D'ALGORITHME ET PROGRAMMATION

Q1) Qu'est-ce un algorithme ?

R/ Un algorithme est une séquence d'étapes de calcul qui transforme une entrée en une sortie. Donc un algorithme est une procédure de calcul qui prend en entrée une valeur, ou un ensemble de valeurs, et qui donne en sortie une valeur ou un ensemble de valeurs.

Q2) Qu'est-ce un algorithme efficace ?

R. L'efficacité d'un algorithme est son temps minimal d'exécution, on dit que son temps d'exécution est constant, idéalement l'unité.

Q3) Que pouvez-vous dire à propos de l'efficacité d'un algorithme ?

R. Elle est d'une grande importance et ne dépend pas de la performance de la machine sur lequel s'exécute ledit algorithme, et dans le cours, nous avons vu que le mieux à faire était d'évaluer l'algorithme avec l'approche théorique car ne dépend pas des limites de la machine qui l'exécute. Mais aussi l'algorithme doit être correct, c'est-à-dire produire une bonne sortie pour une bonne entrée

Q4) Citez quelques-unes des techniques de conception d'un algorithme

R. 1°) La méthode de la force brute ;

2°) La méthode gloutonne ;

3°) La méthode du diviser pour régner ;

4°) La méthode probabiliste ;

5°) L'approche par la programmation dynamique.

Q5) Commentons en quelques phrases :

R. 1°) La méthode de la force brute est une approche qui consiste à essayer toutes les solutions possibles.

2°) La méthode de gloutonne consiste à construire une solution de manière incrémentale en optimisant de manière aveugle un critère local ; c'est-à-dire c'est un algorithme qui étape par étape fait le choix d'un optimum local.

3°) La méthode du diviser pour régner, le problème à résoudre est divisé en sous-problèmes semblables au problème initial, mais de taille moindre. Ensuite les sous-problèmes sont résolus de manière récursive et enfin les solutions des sous-problèmes sont combinées pour avoir la solution du problème original.

4°) La méthode probabiliste fait des choix aléatoires au cours de son exécution.

5°) L'approche par programmation dynamique consiste à trouver la solution optimale en faisant la combinaison des solutions optimales d'une série de sous-problèmes qui se chevauchent.

Q6) Qu'est-ce donc ?

R. – le pseudocode est un LDA, ou langage de description, qui permet de donner l'algorithme dans un langage humain, expressif sans respecter la syntaxe d'un quelconque langage de programmation. - les ordinogrammes (organigrammes) : est une représentation graphique des opérations et des décisions qui résument un algorithme. - l'implémentation de l'algorithme dans un langage de programmation bien spécifique (python, C++, java,)

Q7) Pour quelles raisons une équipe de développeurs de logiciels choisit-elle de représenter les algorithmes par :

R. - du pseudocode : pour permettre à chaque sous-groupe de l'équipe de comprendre l'algorithme dans le cas où l'algorithme fera intervenir par exemple des implémentations dans plusieurs langages de programmation différentes. - des ordinogrammes, utilisés pour faciliter l'exécution et la compréhension de l'algorithme. - des bouts de code : si toute l'équipe développe directement autour d'un seul langage de programmation

Q8) R. Il est vrai que nous pouvons développer plusieurs algorithmes dans un ensemble. En d'autres termes, le bon algorithme est celui qui arrive à produire la solution avec le temps d'exécution le plus minimum possible mais aussi sollicite moins la mémoire (bien que le critère sollicitation de la mémoire ne soit pas pris en compte le plus souvent)

Q9) R. L'analyse d'un algorithme consiste à trouver son temps d'exécution et dire s'il est correct, c'est-à-dire si à la bonne entrée correspond la bonne sortie.

Q10) Quelles sont les deux méthodes d'analyse d'un algorithme ?

R. 1°) **Analyse expérimentale**

2°) **Analyse théorique**

Q11) Quels sont les inconvénients de la méthode expérimentale ?

R. Le fait qu'elle dépend ou donne le temps d'exécution d'un algorithme en fonction des paramètres de la machine sur laquelle on effectue l'expérimentation. Or les machines ont toujours des failles

Q12) En quoi consiste la méthode des opérations primitives ?

R. Une opération primitive est une analyse théorique qui consiste à analyser une instruction de bas-niveau (les opérations de base tels que : assigner une valeur à une variable, effectuer une opération arithmétique, comparer deux nombres, sortir d'une méthode) avec un temps d'exécution constant

Q13) Qu'est-ce la complexité d'un algorithme ?

R. La complexité d'un algorithme consiste à nous focaliser sur l'analyse des algorithmes avec la relation entre le temps d'exécution et la taille de l'entrée comme paramètre principal d'analyse.

Q14) En quoi consiste la notation asymptotique ?

R. Elle consiste tout simplement à décrire le temps d'exécution des algorithmes par des notations beaucoup plus claires sous forme des fonctions polynômes, linéaires, quadratiques, logarithmiques.

Q15) Quelles sont les fonctions qui apparaissent le plus souvent lors de l'analyse théorique des algorithmes ?

R. Nous citons :

-fonction constante $f(n) = c$, fonction logarithme $f(n) = \log bn$,

- fonction linéaire $f(n) = n$, fonction $f(n) = n \log_2 n$,
- la fonction quadratique $f(n) = n^2$, fonction cubique $f(n) = n^3$,
- fonction exponentielle $f(n) = b^n$, $b > 0$

Q16) R. le bon algorithme pour résoudre un problème est celui qui arrive à produire la solution avec le temps d'exécution le plus minimum possible mais aussi sollicite moins la mémoire (bien que le critère sollicitation de la mémoire ne soit pas pris en compte le plus souvent)

Q17) R. La taille d'entrée est le nombre d'éléments qui constituent une entrée, par exemple la longueur n d'un tableau à trier, c'est aussi le total de bits nécessaires à la représentation de l'entrée dans la notation binaire habituelle. Si on se réfère du cours d'informatique de Premier graduat, on embrasse la notion sur les additionneurs, qui sont composés des demi-additionneurs. Ici, pour additionner deux entiers décimaux, il faut les convertir en décimaux, et ici chaque décimal s'écrit avec un certain nombre des 1 et des 0, la somme des 1 et 0 est la taille d'entrée.

Q18) R. la récursivité est un processus par lequel une fonction s'appelle elle-même au cours de son exécution

Imaginons que nous avons une base de données, contenant plusieurs électeurs Et notre problème est de trouver l'identité de la personne suspectée. Notre cas le plus favorable est celui où nous tombons dès la première recherche avec notre algorithme sur ladite identité. Le cas le plus défavorable est celui où nous trouvons le résultat voulu à la fin, en dernier lieu de la recherche. Cependant, le cas le plus défavorable est très important car indique le temps maximal que prend nombre l'algorithme.

Q19) R. La récursivité consiste à avoir un cas de base, sur laquelle on reviendra chaque fois. On définit une fonction qui s'appelle elle-même jusqu'à revenir au cas de départ ou de base.

Q20) R. La récursivité binaire est utilisée pour rechercher un objet dans une séquence triée d'objets. Il s'agit d'un des plus importants algorithmes en usage.

Q21) R. un problème récursif doit se représenter de la manière suivante :

- un cas de base

- *une modification de son état pour se ramener au cas de base*
- *s'appeler elle-même Ex : le factoriel, l'exponentiation rapide (la puissance énième),*