



FACULTE POLYTECHNIQUE
DEUXIEME GRADUAT

DEVOIR -2 COURS D'ALGORITHME ET PROGRAMMATION

Fait à Kinshasa , 05-02-2022 par :

Kambale MARUBA Exaucé (2GEI)

Murhabazi Mutuzi Cedric(2GC)

Twite Ilunga(2GEI)

1. Qu'est-ce un algorithme ?

R. l'algorithme est un outil qui permet de résoudre un problème bien spécifique (de calcul, ou d'une autre nature). Il se résume comme une suite d'étapes claires jusqu'à l'obtention de la solution d'un problème.

2. Qu'est-ce un algorithme efficace ?

R. un algorithme efficace est celui qui arrive à la solution avec un temps d'exécution minimum, on dit que son temps d'exécution est constant, idéalement l'unité.

3. Que pouvez-vous dire à propos de l'efficacité d'un algorithme ?

R. L'efficacité d'un algorithme est très importante et ne dépend pas de la performance de la machine sur lequel s'exécute celui-ci, et dans le cours 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

4. Citez quelques-unes des techniques de conception d'un algorithme

R. on a :

- la méthode de la force brute
- la méthode gloutonne
- la méthode du diviser pour régner
- la méthode probabiliste

Et l'approche par la programmation dynamique

5. Commentons en quelques phrases :

R. – la force brute : notre algorithme n'est pas efficace, puisqu'essaye toutes les possibilités. Si nous avons par exemple des tableaux avec n éléments, faire des branchements avec des if...else...notre algorithme aura un temps d'exécution pas constant.

- la méthode gloutonne est dans plusieurs contextes mieux que celle de la force brute. Car elle évolue en recherchant la solution optimum.

- la méthode de diviser pour régner est beaucoup plus intéressant pour des gros problèmes car elle divise le problème en plusieurs sous problèmes et les résout d'une manière récursive. On divise, on résout les sous problèmes puis on les combine.

- la méthode probabiliste est aléatoire et dans le contexte restreint des algorithmes, il n'est pas trop utilisé car le domaine des aléatoires on ne sait pas exactement donner le temps d'exécution

- la méthode de programmation dynamique est beaucoup mieux, car en divisant le problème en sous problèmes, recherche les solutions optimales des sous problèmes et en fait la solution de l'ensemble.

6. 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,...)

7. 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 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

8. R. Il est vrai qu'on peut développer plusieurs algorithmes dans un ensemble. En bref, 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 n'est pas pris en compte le plus souvent)

9. 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.

10. Quelles sont les deux méthodes d'analyse d'un algorithme ?

R. – l'analyse théorique

- l'analyse expérimentale

11. 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

12. En quoi consiste la méthode des opérations primitives ?

R. Cette méthode se base sur l'assignation des valeurs aux variables, effectuer des opérations arithmétiques, comparer des nombres, indexer un tableau, suivre la référence d'un objet, sortir une méthode

13. 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.

14. 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, ...

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

R. fonction constante $f(n) = c$, fonction logarithme $f(n) = \log_b n$, 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 $f(n) = b^n, b > 0$

16. 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 n'est pas pris en compte le plus souvent)

17. 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.

18. R. Imaginons que nous avons une longue liste de tous les étudiants de la faculté Polytechnique de l'UNIKIN. Et notre problème est de trouver le nom de l'étudiant Maruba. Notre cas le plus favorable est celui où nous tombons dès la première recherche avec notre algorithme sur ce nom Maruba. Le cas le plus défavorable est celui où nous trouvons le nom Maruba à la fin, en dernier lieu de la recherche. Le cas le plus défavorable est très important car indique le temps maximal que prend nombre l'algorithme.

19. 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.

20. 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.

21. R. un problème récursif doit s'énoncer 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 n ieme),...