UNIVERSITE DE KINSHASA
FACULTÉ POLYTECHNIQUE

TRAVAIL DIRIGÉ D'ALGORITHMIQUE ET PROGRAMMATION

Présenté par : MVALA MASASU Eugène(2GC)
NZONGBO DUNGUSE Achille(2GC)

Année Académique 2021-2022

Fiche 2 : **Les FONDEMENTS MATHÉMATIQUE DE L'ALGORITHMIQUE ET L'ÉVOLUTION DES ALGORITHMES**

Répondre aux questions suivants(révision de la matière)

1) Qu'est-ce l'algorithme?
2) Qu'est-ce qu'un programme efficace?
3) Que pouvez-vous dire à propos de l'efficacité d'un algorithme?
4) citer quelques-unes des techniques de conception d'un algorithme?
5) commentez en quelques phrases les techniques de conception des algorithmes suivants: la méthode de la force brute, la méthode de gloutonne, la methode du diviser pour mieux régner, la méthode probabiliste, la méthode de la programmation dynamique.
6) qu'est-ce qu'un pseudo-code? qu'est-ce qu'un organigramme? De quelle autre façon peut-on présenter un algorithme?
7) Pour quelles raisons une équipe de développeurs de logiciels choisit-elle de représenter les algorithmes par du pseudo-code, des organigrammes ou des bouts de code.
8) En général, pour un problème donné, on peut développer plusieurs algorithmes. Comment identifier le meilleur algorithme de cet ensemble?
9) en quoi consiste l'analyse d'un algorithme?
10) Quelles sont les deux méthodes d'analyse d'un algorithme?
11) Quels sont les inconvénients de la methode expérimentale?
12) en quoi consiste la methode des opérations primitives?
13) Qu'est-ce que la complexité d'un algorithme?
14) en quoi consiste la notation asymptotique?
15) Quelles sont les fonctions qui apparaissent le plus lors de l'analyse théorique des algorithmes?
16) Quel est l'algorithme le plus efficace parmi un ensemble d'algorithmes permettant de résoudre un problème?
17) Pour évaluer expérimentalement un algorithme, on doit l'implémenter et lui fournir des entrées différentes questions de mesurer le temps d'exécution correspondant à chaque entrée. c'est en dessinant la courbe du temps d'exécutions en fonction de la taille de l'entrée que l'on peut identifier la fonction correspondant à l'évolution du temps d'exécution en fonction de la taille d'entrée. La notion de la taille d'une entrée est très importante. Pourriez-vous la définir en quelques mots et donner quelques exemples de taille d'entrée pour des problèmes simples.
18) Dans l'analyse d'un algorithme on distingue généralement le cas le plus défavorable, le cas le plus favorable et le cas moyen(probabiliste). Explique en quoi consiste chaque cas. Pourquoi le cas le plus défavorable a une importance particulière?
19) définir en quelques mots le concept de récursivité.
20) En quoi consistent la récursivité linéaire, la récursivité binaire et la récursivité multiple?

RÉPONSES

1) Un algorithme est un outil permettant de résoudre un problème de calcul bien spécifié. Un algorithme est une procédure, étape par étape, permettant de résoudre un problème dans un intervalle de temps fini.

2) L'efficacité ici est synonyme de temps d'exécution minimal.bon algorithme, mieux l'algorithme efficace est celui qui résout le problème en un temps minimal.

3) L'efficacité ici est synonyme de temps d'exécution minimal.

4) La méthode de la force brute;
   La méthode gloutonne;
   La methode du diviser pour mieux régner;
   La méthode probabiliste;
   La méthode de la programmation dynamique.

5) **La méthode de la force brute** est une approche qui consiste à essayer toutes les solutions possibles. Par exemple, pour trouver le maximum d'un ensemble de nombres, on essaye tous les nombres jusqu'à ce que l'on obtienne le maximum.

   Dans la méthode gloutonne, on construit une solution de mani`ere incr´ementale en optimisant de mani`ere aveugle un crit`ere local. Un algorithme glouton est un algorithme qui, étape par étape, fait le choix d'un optimum local. Dans certains cas, cette approche permet d'arriver à un optimum global, mais dans le cas général c'est une heuristique.

   Dans l'approche 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. Le paradigme du diviser pour régner implique trois étapes à chaque niveau de récursivité ,à savoir : diviser, régner et combiner.

   La méthode probabiliste fait appel aux nombres aléatoires. Un algorithme est dit probabiliste lorsqu'il fait des choix aléatoires au cours de son exécution. Un tel algorithme fait appel à un ou plusieurs générateurs de nombres aĺeatoires.

   L'approche par programmation dynamique, la solution optimale est trouvée en combinant des solutions optimales d'une série de sous-problèmes qui se chevauchent.

6) **Le pseudo-code** appelé également Langage de Description d'Algorithmes (LDA) est une façon de décrire un algorithme sans référence à un langage de programmation particulier. L'écriture du pseudo-code permet souvent de bien prendre toute la mesure de la difficulté de la mise en œuvre de l'algorithme et de développer une démarche structurée dans la conception de celui-ci. Le pseudo-code est destiné aux

humains, même s'il existe des langages de spécifications qui au départ du pseudo-code et des diagrammes de toutes sortes peuvent générer du code.

Un organigramme, également appelé algorigramme, logigramme ou ordinogramme est une représentation graphique normalisée des opérations et des décisions effectuées par un ordinateur.

On peut aussi présenter un algorithme par une **implémentation dans un langage de programmation.**

7) L'écriture **du pseudo-code** permet souvent de bien prendre toute la mesure de la difficulté de la mise en œuvre de l'algorithme et de développer une démarche structur´ee dans la conception de celui-ci.

8) Le bon algorithme (l'algorithme efficace) est celui qui résout le problème en un temps minimal.

9) l'analyse des algorithmes avec la relation entre le temps d'exécution et la taille de l'entrée comme paramètre principal d'analyse. Dans la littérature on parle de l'analyse de la complexité d'un algorithme. Dans ce contexte, tout algorithme peut être analysé, soit expérimentalement, soit théoriquement. Le bon algorithme, mieux l'algorithme efficace, est celui qui résout le problème en un temps minimal.

10) **L' analyse expérimentale** et **l'analyse théorique**.

11) - Les expériences ne peuvent être faites que sur un nombre limité d'entrées (d'autres entrées pouvant se révéler importantes sont laissées de côté) ;

- Il est difficile de comparer les temps d'exécution expérimentaux de deux algorithmes sauf si les expériences ont été menées sur les mêmes environnements (Hardware et Software) ;

- On est obligé d'implémenter et d'exécuter un algorithme en vue d'étudier ses performances. Cette dernière limitation est celle qui requiert le plus de temps lors d'une étude expérimentale d'un algorithme.

12) En fait, une opération primitive correspond à une instruction de bas-niveau avec un temps d'exécution constant. Pour déterminer le temps d'exécution d'un algorithme, il suffit de compter le nombre d'opérations primitives exécutées et pour chaque opération primitive de multiplier ce nombre par son temps d'exécution constant.

13) On parle de complexité d'un algorithme, la mesure de la longueur de ses traces d'exécution en fonction de ses paramètres d'entrée.

14) **La notation asymptotique** permet de décrire les temps d'exécution des algorithmes.

15) – assigner une valeur à une variable ; - effectuer une opération arithmétique (exemple : additionner deux nombres) ; - comparer deux nombres ; - indexer un tableau ; – suivre la référence d'un objet ; - sortir d'une méthode.

16) On considère généralement qu'un algorithme est plus efficace qu'un autre si le temps d'exécution de son cas le plus défavorable à un ordre de grandeur inférieur.

17) La taille d'entrée est le nombre d'éléments constituant l'entrée, par exemple la longueur n du tableau à trier, la multiplication de deux entiers.

18) Une analyse basée sur **le cas moyen** exige que nous puissions évaluer les temps d'exécution d'une distribution d'entr´ees, ce qui implique des calculs probabilistes compliqués

    L'analyse basée sur **le plus mauvais cas(défavorable)** exige que l'on puisse identifier l'entrée correspondant au plus mauvais cas et cela est simple à faire. Cette approche mène le plus souvent au meilleur algorithme.On consid`ere généralement qu'un algorithme est plus efficace qu'un autre si le temps d'exécution de son cas le plus défavorable à un ordre de grandeur inférieur

19) **La récursivité** est un processus par lequel une fonction s'appelle elle-même au cours de son exécution.

20) On parle **de la récursivité linéaire** lorsqu'une fonction récursive est conçue pour que chaque invocation du corps fasse au plus un nouvel appel récursif, on parle alors de récursivité linéaire.

    On parle **de la récursivité binaire** lorsqu'une fonction effectue deux appels récursifs.

    On parle **de la récursivité multiple** lorsqu'une fonction peut effectuer plus de deux appels récursifs.