

## Projet II

### 1) Qu'est-ce qu'un algorithme ?

R/ Un algorithme est une méthode de calcul prenant en entrée une valeur, ou un ensemble de valeurs, et qui en ressort une autre dans le but de résoudre un problème ou accomplir une tâche particulière dans un intervalle de temps fini. C'est un ensemble de séquence, d'étapes de calcul qui transforme une entrée en une sortie. Les algorithmes sont largement utilisés dans les domaines de l'informatique, de la science des données et de l'intelligence artificielle.

### 2) Qu'est-ce qu'un algorithme efficace ?

R/ Un algorithme efficace est un algorithme qui possède un temps de calcul court et une faible utilisation de la mémoire. Il est capable de résoudre rapidement et de manière fiable des problèmes complexes en utilisant le moins de ressources possibles.

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

R/ L'efficacité d'un algorithme peut être mesurée de plusieurs manières, telles que la vitesse d'exécution, l'utilisation de la mémoire, le nombre d'opérations effectuées, etc. Il est important de choisir la mesure appropriée en fonction du contexte et des objectifs spécifiques.

### 4) Citer quelques-unes des techniques de conception d'un algorithme ?

R/ On a donc les techniques de conception d'un algorithme :

- La méthode de la force brute
- La méthode gloutonne
- La méthode de diviser pour mieux régner
- La méthode probabiliste
- L'approche de la programmation dynamique

### 5) Commentez en quelques phrases les techniques de conception des algorithmes suivantes : la méthode de la force brute, la méthode gloutonne, la méthode du diviser pour régner, la méthode probabiliste, la méthode de la programmation dynamique.

R/ La méthode de la force brute consiste à tester toutes les possibilités jusqu'à ce que la solution optimale soit trouvée, ce qui peut être inefficace pour des problèmes de grande envergure.

La méthode gloutonne consiste à construire une solution de manière incrémentale en optimisant de façon aveugle un critère local. Un algorithme utilisant cette méthode, étape par étape, fait le choix optimum.

La méthode du diviser pour régner consiste à diviser un grand problème en sous-problèmes plus petits et plus simples, puis à résoudre ces derniers indépendamment avant de les combiner pour obtenir la solution du problème d'origine. On peut le résumer en Diviser-Régner-Combiner.

La méthode probabiliste utilise des approches statistiques pour résoudre des problèmes, souvent en utilisant des algorithmes de simulation ou de calcul de probabilités.

La méthode de la programmation dynamique consiste à conserver les résultats intermédiaires dans une table pour éviter de recalculer les mêmes sous-problèmes plusieurs fois, ce qui peut améliorer considérablement les performances de certains algorithmes.

- 6) Qu'est-ce qu'un pseudo-code? Qu'est-ce qu'un ordinogramme? De quelle autre façon peut-on présenter un algorithme?

R/

- Un pseudo-code ou LDA est une façon de décrire un algorithme sous forme de langage informel. Il est conçu pour être facile à comprendre et à lire, et est utilisé pour décrire la logique d'un algorithme.
- Un organigramme ou algorithme est une représentation graphique normalisée des opérations et des décisions effectuées par un algorithme. Il est composé de différents symboles et de lignes qui représentent les différentes étapes d'un algorithme. Les symboles représentent des opérations et des décisions, et les lignes représentent le flux de contrôle. Il est utilisé pour représenter un algorithme de manière claire et concise.
- On peut représenter un algorithme une implémentation directe en langage de programmation

- 7) Pour quelles raisons une équipe de développeurs de logiciels choisit-elle de représenter les algorithmes par du pseudocode, des organigrammes ou des bouts de code.

R/ Ils choisissent un modèle ou un autre pour des raisons de contexte, objectif de la démarche, compétence de l'équipe et les attentes du public cible. ces méthodes offrent une meilleure compréhension des étapes et des procédures nécessaires pour atteindre un objectif donné.

- 8) En général pour un problème donné, on peut développer plusieurs algorithmes. Comment identifier le meilleur algorithme de cet ensemble ?

R/ En faisant l'analyse théorique ou l'analyse expérimentale de chaque algorithme et voir celui qui le moins de temps dans son exécution. Comparez ensuite les performances des différents algorithmes et choisissez celui qui obtient les résultats les plus précis et qui est le plus efficace.

9) En quoi consiste l'analyse d'un algorithme ?

R/ L'analyse d'un algorithme consiste en l'analyse de son pseudocode.

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

R/ L'analyse théorique et l'analyse expérimentale (après implémentation).

11) Quels sont les inconvénients de la méthode expérimentale ?

- R/ - Les expériences ne peuvent être faites que sur un nombre limité d'entrées
- Il est difficile de comparer les temps d'exécutions expérimentaux de 2 algorithmes sauf si les expériences ont été menées sur les mêmes environnements (Hardware et Software)
  - On est obligé d'implémenter et d'exécuter l'algorithme en vue d'étudier ses performances.

12) En quoi consiste la méthode des opérations primitives ?

R/ En ce qui concerne les opérations primitives, ce sont des opérations de niveau inférieur ayant un temps d'exécution constant. C'est donc l'analyse théorique.

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

R/ La complexité d'un algorithme est une mesure du temps ou de la mémoire nécessaires pour exécuter un algorithme en fonction de la taille de l'entrée. Il existe deux types de complexité : la complexité en temps (complexité temporelle) et la complexité en espace (complexité mémoire). La complexité temporelle mesure le nombre d'opérations effectuées par un algorithme pour résoudre un problème donné, tandis que la complexité mémoire mesure la quantité de mémoire nécessaire pour exécuter l'algorithme.

14) En quoi consiste la notation asymptotique ?

R/ La notation asymptotique d'un algorithme consiste à évaluer la performance d'un algorithme en termes de sa complexité en temps ou en espace.

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

R/ Nous avons : les fonctions constantes, les fonctions logarithmiques, les fonctions linéaires, les fonctions nlogn, les fonctions quadratiques, les fonctions cubiques, les fonctions polynômes ( $n \geq 4$ ) et les fonctions exponentielles.

16) Quel est l'algorithme le plus efficace parmi un ensemble d'algorithmes permettant de résoudre un problème

R/ C'est l'algorithme avec le moins de temps d'exécution.

17) Pour évaluer expérimentalement un algorithme on doit l'implémenter et lui fournir des entrées différentes question de mesurer le temps d'exécution correspondant à chaque entrée. C'est en dessinant la courbe du temps d'exécution 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.

R/ Pour évaluer un algorithme expérimentalement, nous devons le mettre en œuvre et lui fournir des entrées différentes afin de mesurer le temps d'exécution correspondant à chaque entrée. En traçant la courbe du temps d'exécution en fonction de la taille de l'entrée, nous pouvons identifier la fonction qui décrit l'évolution du temps d'exécution en fonction de la taille de l'entrée. On a comme cas le plus parlant de tri par insertion, plus grande est l'entrée et plus long sera le tri ; la recherche de données dans la recherche binaire, etc.

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). Expliquez, en quoi consiste chaque cas. Pourquoi le cas le plus défavorable a une importance particulière ?

R/ Le cas le plus favorable est le cas optimum. C'est-à-dire qu'on ne peut pas résoudre le problème en moins de temps dans ce cas.

Le cas le plus défavorable est la pire des situations, c'est-à-dire qu'on ne peut pas résoudre le problème dans les temps impartis que dans ce cas. Il permet tout de même de déterminer le temps maximal nécessaire pour résoudre le problème.

Le cas moyen (probabiliste) est celui où le temps nécessaire pour résoudre le problème est aléatoire.

19) Définir en quelques mots le concept de récursivité.

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

20) En quoi consistent la récursivité linéaire, la récursivité binaire et la récursivité multiple

MBONGA MABIALA GLOIRE  
DZAPILI JOSEPH  
BOLOCHI KASIMA PIERRE

R/ - La récursivité linéaire consiste en ce que pour chaque invocation du corps, l'algorithme fasse au plus un nouvel récursif. (Récursivité type factoriel)

- La récursivité binaire consiste en ce que la fonction fasse deux appels récursifs
- La récursivité multiple consiste en ce que la fonction fasse plusieurs appels récursifs au moins plus de 2 fois.

21) De quelle façon un problème récursif doit-il pouvoir se définir ? Donnez un exemple

R / Un problème récursif se définit en termes d'appel récursif. Cela signifie qu'une tâche est divisée en sous-tâches qui mènent à la solution finale. Un exemple est le tri par sélection.