



Final Exam, Algorithms 2019-2020

- You are only allowed to have a handwritten A4 page written on both sides.
- Communication, calculators, cell phones, computers, etc... are not allowed.
- Your explanations should be clear enough and in sufficient detail that a fellow student can understand them. In particular, do not only give pseudocode without explanations. A good guideline is that a description of an algorithm should be such that a fellow student can easily implement the algorithm following the description.
- Attached at the end of the exam is a French translation.
- **Do not touch until the start of the exam.**

Good luck!

Name: _____

N° Sciper: _____

Problem 1	Problem 2	Problem 3	Problem 4	Problem 5	Problem 6
/ 23 points	/ 16 points	/ 18 points	/ 16 points	/ 17 points	/ 10 points

Total / 100

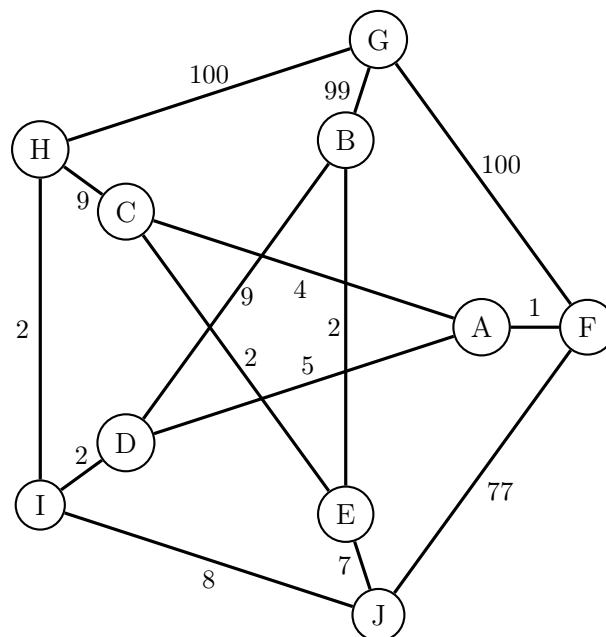
- 1 (23 pts) **Basic questions.** This problem consists of five subproblems (1a-1e) for which you **do not** need to motivate your answers.

1a (5 pts) Arrange the following functions in increasing order according to asymptotic growth.

- A $\log(n!)$
- B $(\log n)^{\log n}$
- C 2^n
- D The worst case running time of BUILD-MAX-HEAP on an array of size n .
- E The worst case running time of Strassen's algorithm for multiplying two $n \times n$ matrices.
- F The worst case running time of QUICKSORT on an array of size n .

Solution:

- 1b (4 pts) In the graph shown in the figure below, what order are the nodes reached in if we run Prim's algorithm from A?



Solution:

- 1c** (6 pts, 1 pt for each correct pair) Pair up each of the recurrence relations on the left side with the correct asymptotic growth function on the right side (assuming that $T(1) = 1$). For example the recurrence $T(n) = T(n-1) + 1$ has asymptotic growth $\Theta(n)$.

Solution:

$T(n) = T(n-1) + 1$	$\Theta(n^2)$
$T(n) = 4T(n/3) + n^2$	$\Theta(n)$
$T(n) = 4T(n-1) + 1$	$\Theta(n^3 \log n)$
$T(n) = 10T(n/10) + n^3 \log n$	$\Theta(4^n)$
$T(n) = 8T(n/2) + 1$	$\Theta(n^4 \log n)$
$T(n) = 4T(\sqrt{n}) + \log n$	$\Theta(\log^2 n)$
$T(n) = 16T(n/2) + n^4$	$\Theta(n^3)$

- 1d** (4 pts) Consider the recurrence $T(n) = 2T(\alpha n) + 5T(\beta n) + n^2$, $T(1) = \Theta(1)$, for some constants $\alpha, \beta \in (0, 1)$ that satisfy $2\alpha + 5\beta = 1$. Give a tight asymptotic bound for $T(n)$. You do not need to motivate your answer.

Solution:

The answer is $T(n) = \Theta(\text{_____})$

- 1e** (4 pts) Consider the recurrence $T(n) = 2T(\alpha n) + 5T(\beta n) + n^2$, $T(1) = \Theta(1)$, for some constants $\alpha, \beta \in (0, 1)$ that satisfy $2\alpha^2 + 5\beta^2 = 1$. Give a tight asymptotic bound for $T(n)$. You do not need to motivate your answer.

Solution:

The answer is $T(n) = \Theta(\text{_____})$

- 2 (16 pts) Binary trees.** In this problem you are given a rooted binary tree $T = (V, E)$ with n nodes, and your task is to design an efficient algorithm for finding a pair of nodes $a, b \in V$ that are at the largest possible distance from each other in T (the distance between a and b in T is the length of the path connecting a to b in T that does not have repeated vertices). For full credit your algorithm should run in $O(n)$ time.

- **Input:** A rooted binary tree $T = (V, E)$.
- **Output:** A pair of nodes $a, b \in V$ at maximum possible distance from each other in T .

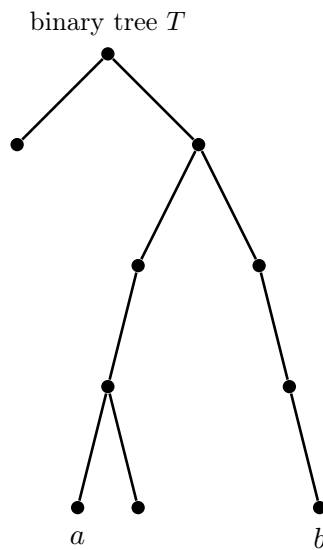


Figure 1. A binary tree T with a pair of nodes a, b at maximum possible distance in T .

Design an algorithm for this problem and analyze its runtime.

Solution:

Continuation of the solution to 2:

- 3 (18 pts) **Fair partitioning.** In this problem you are given n positive integers $(w_1, \dots, w_n)$ with total sum $W = \sum_{i=1}^n w_i$. Your task is to determine whether it is possible to partition these integers into three disjoint subsets such that the sum of elements in every subset is exactly $W/3$.

- **Input:** An integer n and a sequence of positive integers $(w_1, \dots, w_n)$ whose total sum is $W = \sum_{i=1}^n w_i$.
- **Output:** YES if it is possible to find a partition I_1, I_2, I_3 of the integers between 1 and n such that for every $j = 1, 2, 3$ one has $\sum_{i \in I_j} w_i = W/3$, and NO otherwise.

For example, suppose that the input sequence is $w_1 = 5, w_2 = 6, w_3 = 7, w_4 = 1, w_5 = 1, w_6 = 1$. Then such a partitioning exists: we let $I_1 = \{1, 4, 5\}, I_2 = \{2, 6\}, I_3 = \{3\}$. On the other hand, if the input sequence is $w_1 = 5, w_2 = 3, w_3 = 7, w_4 = 8, w_5 = 4, w_6 = 3$, then no such partitioning exists.

Design and analyze a dynamic programming solution to the problem with runtime polynomial in n and W .

- 3a For $i \in \{1, 2, \dots, n\}$ and two integers $x, y \in \{0, 1, \dots, W\}$ let $d(i, x, y) = 1$ if it is possible to partition the integers $w_1, w_2, \dots, w_i$ (the first i integers in our sequence) into three sets whose sums are x, y and $\sum_{j=1}^i w_j - (x+y)$ respectively, and $d(i, x, y) = 0$ otherwise. Design a recurrence relation for $d(i, x, y)$.

Solution:

3b Analyze the runtime of a bottom up implementation of your recurrence in **3a**.

Solution:

- 4 (16 pts) **Connectivity.** In this problem your task is to design cost effective infrastructure for a country consisting of n cities represented by vertices of a graph $G = (V, E)$. For every pair of cities $a, b \in V$ such that $(a, b) \in E$ you can build a road connecting the cities at cost w_{ab} . Also, for every city $a \in V$ you can build an airport at a at cost c_a . Any two cities that have an airport can reach each other. Your task is to determine which roads should be built and which airports should be constructed to ensure connectivity while minimizing cost.

Input: Connected undirected graph $G = (V, E)$ with non-negative edge weights $w : E \rightarrow \mathbb{R}_+$, for every $a \in V$ the cost c_a of building an airport at a .

Output: Minimum cost collection of roads and/or airports to build that ensures connectivity of all cities in V .

In this problem you are expected to (a) design an efficient algorithm and (b) analyze its runtime.

Solution:

Continuation of the solution to 4:

- 5 (17 pts) **Path finding.** In this problem you are given an undirected graph $G = (V, E)$ and three distinct vertices $x, y, z \in V$. Your task is to find a simple path from x to y in G that goes through the vertex z if such a path exists, and output **NONE** otherwise. Recall that a path is simple if it does not have repeated vertices. See Fig. 2 below for an example.

Input: An undirected graph $G = (V, E)$ together with three distinct vertices $x, y, z \in V$.

Output: A simple path from x to y in G that goes through z if such a path exists, and **NONE** otherwise.

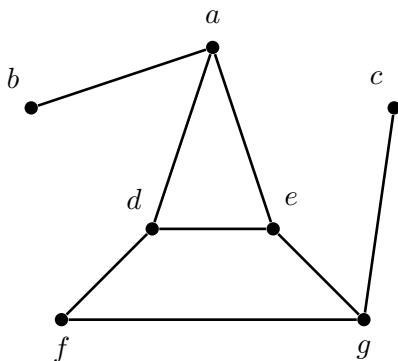


Figure 2. In the graph G above there exists a simple path from b to c that goes through e , but there does not exist a simple path from a to f that goes through b .

Design and analyze an algorithm for the problem. Your algorithm should run in time polynomial in $|V|$. *Hint: modify the graph and use max-flow.*

In this problem you are required to (a) describe an efficient algorithm and (b) analyze its runtime.

Solution:

Continuation of the solution to 5:

- 6 (10 pts) **Searching in a 2d array.** In this problem you are given an $n \times n$ array whose columns and rows contain numbers in increasing order, and a number x (see example below). Your task is to design an algorithm that locates x in the array or concludes that it is not there. For full credit your algorithm should run in $O(n)$ time.

1	2	3	5	7
3	3	4	5	8
3	4	5	5	8
4	5	6	8	10
6	7	7	9	11

Figure 3. An example 5×5 array A whose rows and columns contain numbers in increasing order.

- **Input:** Integers n and x , as well as an $n \times n$ array A of integers such that all rows of A are monotone increasing, and all columns are monotone increasing.
- **Output:** Integers $1 \leq i, j \leq n$ such that $A_{i,j} = x$ if x is in A , and $\perp$ otherwise.

Design an algorithm for this problem and analyze its runtime.

Solution:

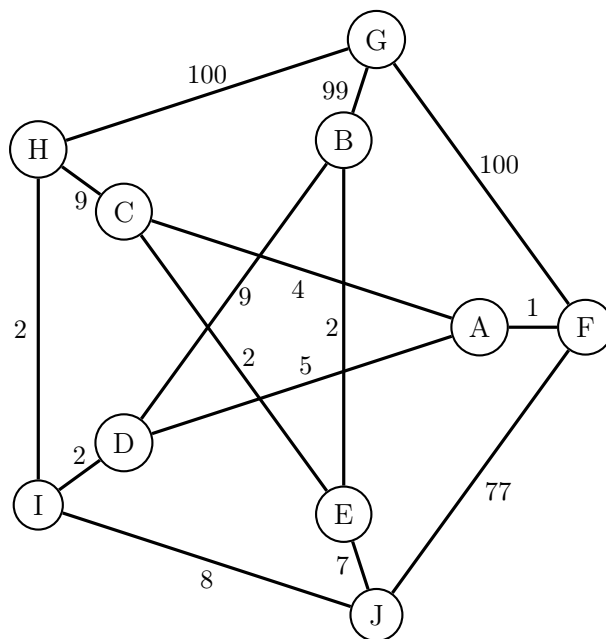


Examen Final, Algorithmique 2019-2020

- Matériel autorisé: une page A4 écrite à la main sur les 2 côtés.
- Communication, calculatrices, téléphones portables, ordinateurs, etc... sont interdits.
- Vos explications doivent être suffisamment claires et détaillées pour qu'un autre étudiant puisse les comprendre. En particulier, ne donnez pas de pseudocode tout seul sans explication. Une bonne indication est qu'un autre étudiant doit pouvoir implémenter facilement l'algorithme en suivant la description.
- Vous êtes autorisé à utiliser des algorithmes traités en classe sans besoin de prouver à nouveau leurs propriétés.
- **Ne touchez pas l'épreuve avant le commencement de l'examen.**

Bonne chance!

- 1 (23 pts) **Questions basiques.** Ce problème se compose de cinq sous-problèmes (1a-1e) pour lesquels vous n'avez pas besoin de justifier vos réponses.
- 1a (5 pts) Ordonnez les fonctions suivantes par ordre croissant de la croissance asymptotique.
- A $\log(n!)$
 - B $(\log n)^{\log n}$
 - C 2^n
 - D Le temps d'exécution dans le pire des cas de BUILD-MAX-HEAP dans un tableau de taille n .
 - E Le temps d'exécution dans le pire des cas de l'algorithme de Strassen pour multiplier deux matrices $n \times n$.
 - F Le temps d'exécution dans le pire des cas de QUICKSORT dans un tableau de taille n .
- 1b (4 pts) Dans le graphe de la figure ci-dessous, dans quel ordre atteint-on les nœuds si on exécute l'algorithme de Prim depuis A?
- 1c (6 pts, 1 pt pour chaque paire correcte) Associez chacune des relations de récurrence de la colonne de gauche à la bonne fonction de croissance asymptotique de la colonne de droite (en supposant que $T(1) = 1$). Par exemple, la récurrence $T(n) = T(n - 1) + 1$ a une croissance asymptotique $\Theta(n)$.



$T(n) = T(n - 1) + 1$	$\Theta(n^2)$
$T(n) = 4T(n/3) + n^2$	$\Theta(n)$
$T(n) = 4T(n - 1) + 1$	$\Theta(n^3 \log n)$
$T(n) = 10T(n/10) + n^3 \log n$	$\Theta(4^n)$
$T(n) = 8T(n/2) + 1$	$\Theta(n^4 \log n)$
$T(n) = 4T(\sqrt{n}) + \log n$	$\Theta(\log^2 n)$
$T(n) = 16T(n/2) + n^4$	$\Theta(n^3)$

- 1d** (4 pts) Considérez la récurrence $T(n) = 2T(\alpha n) + 5T(\beta n) + n^2$, $T(1) = \Theta(1)$, pour des constantes $\alpha, \beta \in (0, 1)$ qui vérifient $2\alpha + 5\beta = 1$. Donnez une borne asymptotique serrée pour $T(n)$. Vous n'avez pas à justifier votre réponse.
- 1e** (4 pts) Considérez la récurrence $T(n) = 2T(\alpha n) + 5T(\beta n) + n^2$, $T(1) = \Theta(1)$, pour des constantes $\alpha, \beta \in (0, 1)$ qui vérifient $2\alpha^2 + 5\beta^2 = 1$. Donnez une borne asymptotique serrée pour $T(n)$. Vous n'avez pas à justifier votre réponse.

- 2** (16 pts) **Arbres binaires.** Dans ce problème, on vous donne un arbre binaire enraciné $T = (V, E)$ à n nœuds, et votre tâche consiste à concevoir un algorithme efficace pour trouver une paire de nœuds $a, b \in V$ qui soient à la plus grande distance possible l'un de l'autre dans T (la distance entre a et b dans T est la longueur du chemin reliant a et b dans T qui n'a pas de sommets répétés). Pour obtenir la totalité des points à ce problème, votre algorithme devra s'exécuter en un temps $O(n)$.

- **Entrée:** Un arbre binaire $T = (V, E)$;
- **Sortie:** Une paire de nœuds $a, b \in V$ à la plus grande distance possible l'un de l'autre dans T .

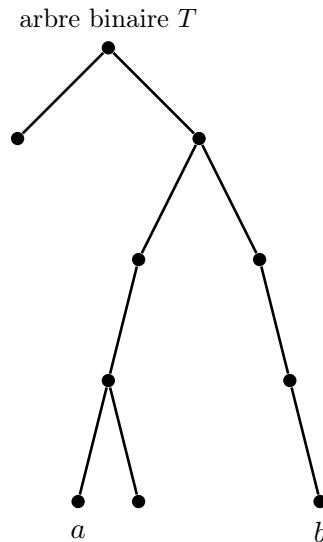


Figure 1. Un arbre binaire T avec une paire de nœuds a et b à la distance maximum possible dans T .

Concevez un algorithme pour ce problème et analysez son temps d'exécution.

- 3 (18 pts) Partitionnement équitable.** Dans ce problème, on vous donne n entiers positifs $(w_1, \dots, w_n)$ dont la somme totale est $W = \sum_{i=1}^n w_i$. Votre tâche consiste à déterminer s'il est possible de partitionner ces entiers en trois sous-ensembles disjoints tels que la somme des éléments de chaque sous-ensemble est exactement $W/3$.

- **Entrée:** un entier n et une séquence d'entiers positifs $(w_1, \dots, w_n)$ dont la somme totale est $W = \sum_{i=1}^n w_i$;
- **Sortie:** YES s'il est possible de trouver une partition I_1, I_2, I_3 des entiers entre 1 et n telle que pour $j = 1, 2, 3$, on a $\sum_{i \in I_j} w_i = W/3$, et NO si c'est impossible.

Par exemple, supposons que la séquence d'entrée soit $w_1 = 5, w_2 = 6, w_3 = 7, w_4 = 1, w_5 = 1, w_6 = 1$. Alors un tel partitionnement existe: on a $I_1 = \{1, 4, 5\}, I_2 = \{2, 6\}, I_3 = \{3\}$. En revanche, si la séquence d'entrée est $w_1 = 5, w_2 = 3, w_3 = 7, w_4 = 8, w_5 = 4, w_6 = 3$ alors un tel partitionnement n'existe pas.

Concevez et analysez une solution de programmation dynamique au problème avec un temps d'exécution polynomial en n et W .

- 3a** Pour $i \in \{1, 2, \dots, n\}$ et deux entiers $x, y \in \{0, 1, \dots, W\}$, soit $d(i, x, y) = 1$ s'il est possible de partitionner les entiers $w_1, w_2, \dots, w_i$ (les premiers i entiers de notre séquence) en trois ensembles dont les sommes sont x, y et $\sum_{j=1}^i w_j - (x + y)$ respectivement, et $d(i, x, y) = 0$ si c'est impossible. Concevez une relation de récurrence pour $d(i, x, y)$.

3b Analysez le temps d'exécution d'une implémentation bottom-up de la récurrence que vous avez conçue en **3a**.

- 4 (16 pts) Connectivité.** Dans ce problème, votre tâche consiste à concevoir à moindre frais l'infrastructure d'un pays comprenant n villes représentées par les sommets d'un graphe $G = (V, E)$. Pour chaque paire de villes $a, b \in V$ telles que $(a, b) \in E$ vous pouvez construire une route reliant les villes pour un coût w_{ab} . En outre, pour chaque ville $a \in V$ vous pouvez construire un aéroport à a pour un coût c_a . Deux villes pourvues d'aéroports sont accessibles l'une à l'autre. Votre tâche consiste à déterminer quelles routes devraient être construites et quels aéroports devraient être bâtis afin de garantir la connectivité à moindre frais.

Entrée: Un graphe connexe non orienté $G = (V, E)$ avec des poids d'arêtes non négatifs $w : E \rightarrow \mathbb{R}_+$, pour chaque $a \in V$ le coût c_a de la construction d'un aéroport à a ;

Sortie: Collection de routes et/ou aéroports à construire pour garantir la connectivité entre toutes les villes de V à coût minimal.

Dans ce problème vous devez (a) concevoir un algorithme efficace et (b) analyser son temps d'exécution.

- 5 (17 pts) Trouver son chemin.** Dans ce problème, on vous donne un graphe non orienté $G = (V, E)$ et trois sommets distincts $x, y, z \in V$. Votre tâche consiste à trouver un chemin simple de x à y dans G qui passe par le sommet z si un tel chemin existe, et produit en sortie **NONE** s'il n'en existe pas. Rappelez-vous qu'un chemin est simple s'il n'a pas de sommets répétés. Voir exemple en Fig. 2 ci-dessous.

Entrée: Un graphe non orienté $G = (V, E)$ ainsi que trois sommets distincts $x, y, z \in V$.

Sortie: Un chemin simple de x à y dans G qui passe par z si un tel chemin existe, et **NONE** dans le cas contraire.

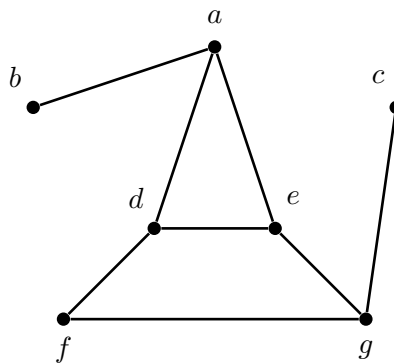


Figure 2. Dans le graphe G ci-dessus, il existe un chemin simple de b à c qui passe par e , mais il n'existe pas de chemin simple de a à f qui passe par b .

Concevez et analysez un algorithme pour ce problème. Votre algorithme doit s'exécuter en un temps polynomial dans $|V|$. *Conseil: modifiez le graphe et utilisez max-flow.*

Dans ce problème vous devez (a) décrire un algorithme efficace et (b) analyser son temps d'exécution.

- 6 (10 pts) **Recherche dans un tableau 2D.** Dans ce problème, on vous donne un tableau $n \times n$ dont les colonnes et les lignes contiennent des nombres dans l'ordre croissant, et un nombre x (voir exemple ci-dessous). Votre tâche consiste à concevoir un algorithme qui trouve x dans le tableau ou conclut qu'il n'est pas dedans. Pour obtenir la totalité des points à ce problème, votre algorithme devra s'exécuter en un temps $O(n)$.

1	2	3	5	7
3	3	4	5	8
3	4	5	5	8
4	5	6	8	10
6	7	7	9	11

Figure 3. Un exemple de tableau 5×5 A dont les lignes et les colonnes contiennent des nombres dans l'ordre croissant.

- **Entrée:** Entiers n et x , ainsi qu'un tableau $n \times n$ A d'entiers tels que toutes les lignes de A sont monotones croissantes, et toutes les colonnes sont monotones croissantes.
- **Sortie:** Entiers $1 \leq i, j \leq n$ tels que $A_{i,j} = x$ si x est dans A , et $\perp$ dans le cas contraire.

Concevez un algorithme pour ce problème et analysez son temps d'exécution.