Graphs :

1 hour 30

No electronic document is allowed. You may have one page of manuscript notes. You can answer in french or english. All answers must be formally explained. All sections are independent. Times are given as informal indications.

Recall that *vertex* means "sommet" in french and *edge* means "arête" in french.

1 Basics (about 30 minutes)

Question 1 *Given a graph G, give the definitions (at most two lines per definition) of*

- *an Eulerian cycle in G ;*
- *an Hamiltonian cycle in G ;*
- *a matching ("couplage" in french) in G.*

Question 2 *Draw a graph with an Eulerian cycle but no Hamiltonian cycle. You must justify (in one sentence) why your graph has an Eulerian cycle but no Hamiltonian cycle.*

Question 3 *Let A and B be any two disjoint connected graphs. Let $u \in V(A)$ be any vertex of A and $v \in V(B)$ be any vertex of B. Let G be the graph obtained from A and B by adding an edge $\{u, v\}$ between u and v. Does G has an Eulerian cycle ? an Hamiltonian cycle ? Justify your answers.*

Question 4 *Give the Berge's theorem that characterizes maximum matching in terms of augmenting paths (for a matching M, you must recall the definition of M-augmenting path).*

Using this theorem, explain why the matching M in the graph G given in Figure 1 is not maximum. Give a maximum matching in G and justify that it is maximum.

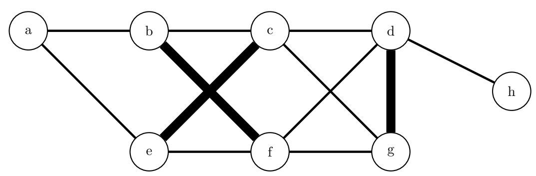

FIGURE 1 – A graph on 8 vertices with a matching $M = \{\{b, f\}, \{e, c\}, \{d, g\}\}$ depicted by bold edges.

Question 5 *In at most two lines, give your definition/intuition of what means to be NP-complete.*

Question 6 *Among the following problems, say which ones can be solved in polynomial time (in which case, you may give their time complexity) and which ones are NP-complete. No justification is required here.*

1. *Given a graph $G = (V, E)$, does it admit an Eulerian cycle ?*
2. *Given a set of tasks (each with some execution time) and p processors, assign each task to some processor in order to minimize the total execution time (once a task is launched in a processor, it must be executed until its end).*
3. *Sort n integers.*
4. *Given a graph $G = (V, E)$ and s and d two vertices of G, compute a shortest path between s and d.*
5. *Given a graph $G = (V, E)$, does it admit an Hamiltonian cycle ?*
6. *Given a graph $G = (V, E)$ with a weight function $w : E \to \mathbb{R}^+$ and $k \geq 0$, does G admit a spanning tree with weight at most k ?*

Question 7 *Consider the problem that, given a set of tasks (each with some execution time) and p processors, assign each task to some processor in order to minimize the total execution time (once a task is launched in a processor, it must be executed until its end). Formally, the inputs are a set of n tasks with execution times $(t_i)_{1 \leq i \leq n}$ and an integer $p \geq 1$ and the goal is to compute a partition $(X_1, \cdots, X_p)$ of $\{1, \cdots, n\}$ that minimizes $\max_{j \leq p} \sum_{i \in X_j} t_i$.*

Give an algorithm that solves this problem optimally or that provides an approximated solution. Precise the performances of your algorithm (time complexity, guarantee on the returned solution).

Your answer must consist of at most about 10 lines and no justification is required.

2 Dynamic programming for the knapsack problem (about 40 minutes)

In the **Knapsack problem**, you have a knapsack with some capacity $C \in \mathbb{N}$ and a set of objects each with some value $v_i \geq 0$ and some weight $c_i \in \mathbb{N}$. You must choose the objects to put in your knapsack in such a way that the total weight of the elements you take does not exceed the capacity C and maximizing the total value of the objects you have taken.

Formally, the inputs of the problem are the capacity C and the n objects $\{x_i = (v_i, c_i)\}_{1 \leq i \leq n}$ and the goal is to find a subset $S \subseteq \{1, \cdots, n\}$ such that $\sum_{i \in S} c_i \leq C$ and its value $\sum_{i \in S} v_i$ is maximum.

Question 8 *For instance, assume the knapsack has capacity $C = 6$ and there are $n = 4$ possible objects : $x_1 = (1, 1), x_2 = (3, 3), x_3 = (3, 3)$ and $x_4 = (4, 5)$. A valid solution is $S = \{1, 4\}$ (i.e., we take the objects x_1 and x_4). What are its weight and its value ? Is it an optimal solution ?*

Question 9 *Give a naive algorithm that, given C and $\{x_i = (v_i, c_i)\}_{1 \leq i \leq n}$, computes an optimal solution. What is the time-complexity of your algorithm ?*

For every $1 \leq s \leq n$ and $0 \leq k \leq C$, we consider the subproblems $\mathcal{P}_{s,k}$ with only objects $(v_i, c_i)_{1 \leq i \leq s}$ and a knapsack of capacity k. Let $OPT[s, k]$ be the maximum value of a solution of the subproblem $\mathcal{P}_{s,k}$. More precisely, let $OPT[s, k]$ be the maximum value that can be obtained from taking objects in the first s objects and fitting in a knapsack of capacity k. That is, $OPT[s, k] = \max_{S' \subseteq \{1, \cdots, s\}} \sum_{j \in S'} v_j$ such that $\sum_{j \in S'} c_j \leq k$.

Question 10 *Give the value $OPT[1, k]$ for every $0 \leq k \leq C$. Give the value $OPT[s, 0]$ for all $1 \leq s \leq n$.*

In what follows, we set $OPT[s,k] = -\infty$ if $k < 0$.

Question 11 *Let $1 \leq s \leq n$ and $1 \leq k \leq C$.*

a Let $S' \subseteq \{1, \cdots, s-1\}$ be any valid solution of $\mathcal{P}_{s-1,k}$. Show that S' is a valid solution of $\mathcal{P}_{s,k}$.

b Recall that c_s is the weight of the s^{th} object. If $c_s \leq k$, let $S' \subseteq \{1, \cdots, s-1\}$ be any valid solution of $\mathcal{P}_{s-1,k-c_s}$. Show that $S' \cup \{s\}$ is a valid solution of $\mathcal{P}_{s,k}$.

Deduce from previous questions that $OPT[s,k] \geq \max\{OPT[s-1,k]; OPT[s-1,k-c_s]+v_s\}$.

Question 12 *Let $1 \leq s \leq n$ and $1 \leq k \leq C$ and $S \subseteq \{1, \cdots, s\}$ be any valid solution of $\mathcal{P}_{s,k}$.*

a If $s \notin S$, show that S is a valid solution of $\mathcal{P}_{s-1,k}$.

b Otherwise (i.e., $s \in S$), show that $S \setminus \{s\}$ is a valid solution of $\mathcal{P}_{s-1,k-c_s}$.

Deduce from previous questions that $OPT[s,k] \leq \max\{OPT[s-1,k]; OPT[s-1,k-c_s]+v_s\}$.

From previous two questions, it follows that, for every $1 \leq s \leq n$ and $1 \leq k \leq C$:

$$OPT[s,k] = \max\{OPT[s-1,k]; OPT[s-1,k-c_s]+v_s\}. \tag{1}$$

Question 13 *Design a dynamic programming algorithm, based on Equation (1), that solves the main problem $\mathcal{P}_{n,C}$. What is its time-complexity (as a function of n and C) ?*

3 Solving a "real life" problem using graphs (about 15 minutes)

Puzzle : We have several pieces (an example is depicted in Figure 2) and the goal is to align all of them in order to create a "wave" (an example of solution is given in Figure 3). Two pieces can be adjacent if they share a vertical side of same height. Note also that any piece x may be swapped (to obtain a symmetrical piece $\bar{x}$ along the vertical axis) [1].

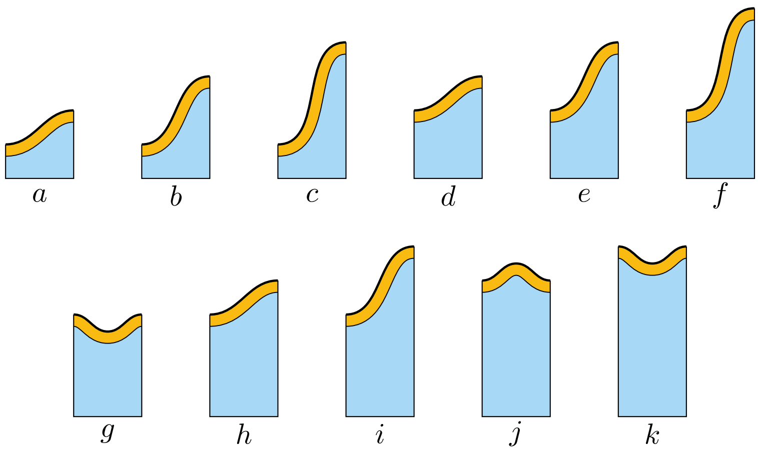

FIGURE 2 – Example of pieces of the puzzle.

Note that each piece is only defined by the heights of its two vertical sides. For instance, the piece a in Fig. 2 can be denoted by $a = (1,2)$ since one of its sides has height 1 and the other 2.

1. Such a puzzle was a test in Koh-Lanta (French version of Survivor) few years ago.

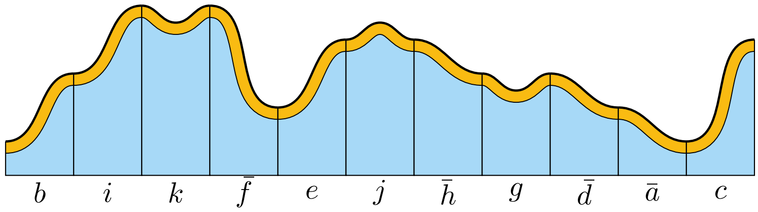

FIGURE 3 – Example of a solution for the pieces given in Figure 2.

The general problem. Given a set of n pieces $P = \{p_i = (a_i, b_i)\}_{1 \le i \le n}$, is there a solution (i.e., a wave aligning the n pieces and respecting the rules) ? If yes, how to find a solution ?

To solve this problem, let us model it by the following graph $G(P)$. The vertices correspond to the heights of the sides of the pieces. That is, there is a vertex labeled x if there is a piece p_i such that $x = a_i$ or $x = b_i$. Moreover, there is an edge between vertices x and y if there is a piece (x, y) (i.e., a piece with one side of height x and the other side of height y). If several pieces (x, y) are similar (i.e., each of these pieces has one side of height x and one side of height y), there as many edges between the vertices x and y (i.e., parallel edges are allowed). Also, each piece (x, x) (i.e., with its two vertical sides of same height x) corresponds to a self-loop in the vertex x. An example of the graph $G(P)$ is depicted in Figure 4.

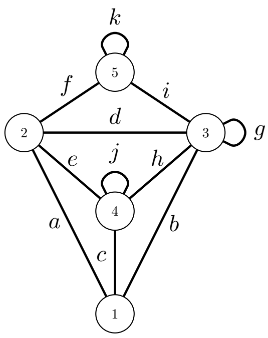

FIGURE 4 – Example of the graph $G(P)$ for the set P of pieces given in Figure 2. Note that the sides of the pieces of this example have exactly 5 different heights that we denote by $1, 2, 3, 4, 5$ and that correspond to the five vertices of $G(P)$. For instance, the piece $a = (1, 2)$ corresponds to the edge between vertices 1 and 2.

Question 14 *Explain how to represent the solution given in Figure 3 in the graph $G(P)$ given in Figure 4.*

Question 15 *Design an algorithm that, given a set of pieces $P = \{p_i = (a_i, b_i)\}_{1 \le i \le n}$, either computes a solution or certifies that no solution exists. Explain your answer (at most 10 lines).*