University Côte d'Azur
Master 2 : Ubinet

Graph Algorithms and Combinatorial Optimization
November, 2024

Final term exam : 1 hour 30

No electronic/manuscript document is allowed. You can answer in french or english. All answers must be carefully justify.

1 Approximation of Vertex Cover (10 minutes)

Let $G = (V, E)$ be a graph. A Vertex Cover of G is a set of vertices that "touch" all edges, i.e., a set $Q \subseteq V$ such that $Q \cap e \neq \emptyset$ for all $e \in E$.

Question 1 *Define a maximal matching in a graph.*

Question 2 *Prove that, if $M \subseteq E$ is a maximal matching, then $V(M)$ (the set of endpoints of the edges in M) is a vertex cover.*

Algorithm 1

Require: A graph $G = (V, E)$.

1: Compute a maximal matching M of G using a greedy algorithm.

2: **Return** $V(M)$

Question 3 *What returns Algorithm 1 ? Why is it an approximation algorithm ?*

2 Approximation of shortest Hamiltonian cycle (40 minutes)

A path $P = (v_1, v_2, \cdots, v_\ell)$ in a graph $G = (V, E)$ is a sequence of distinct vertices such that two consecutive vertices are adjacent in G, i.e., $v_i \neq v_j$ for all $1 \leq i < j \leq \ell$ and $\{v_i, v_{i+1}\} \in E$ for all $1 \leq i < \ell$. This is a cycle if, moreover, $\ell \geq 3$ and $\{v_1, v_\ell\} \in E$.

A walk $W = (v_1, v_2, \cdots, v_\ell)$ in G is a sequence of vertices (not necessarily distinct) such that two consecutive vertices are adjacent in G. A closed walk is a walk $(v_1, v_2, \cdots, v_\ell)$ such that $\ell \geq 3$ and $v_1 = v_\ell$. Note that a path is a walk.

A path/cycle/ (closed) walk is Hamiltonian if it goes through each vertex of the graph.

A spanning tree of a graph G is a subgraph of G that is a tree (connected and acyclic subgraph) and that contains all vertices of G. For instance, a Hamiltonian path in G is a spanning tree of G.

Given a graph $G = (V, E)$ and a non-negative weight function $w : E \to \mathbb{R}^+$ on its edges, the weight of a walk $H = (v_1, v_2, \cdots, v_\ell)$ is the sum of the weight of its edges : $w(H) = \sum_{1 \leq i < \ell} w(\{v_i, v_{i+1}\})$. The weight of a cycle $C = (v_1, v_2, \cdots, v_\ell)$ equals $w(C) = w(v_1, v_\ell) + \sum_{1 \leq i < \ell} w(\{v_i, v_{i+1}\})$. Similarly, the weight $w(T)$ of spanning tree T of G is the sum of the weights of its edges.

Question 4 *Describe an algorithm that computes a spanning tree with minimum weight of a edge-weighted graph. What is its time complexity ?*

Question 5 *Given a spanning tree T with weight W of a graph G, give a Hamiltonian walk of G with weight $2W$.*

(hint : draw an example to catch the intuition.)

Question 6 *Show that, if G admits a Hamiltonian cycle of weight at most W, then it admits a spanning tree of weight at most W.*

From now on, we consider the complete graph K_n on n vertices (i.e., every two vertices are adjacent) with a non-negative weight function $w : E(K_n) \to \mathbb{R}^+$ that satisfies the triangular inequality : that is : for every three vertices $x, y, z \in V(K_n)$, $w(\{x,y\}) \leq w(\{x,z\}) + w(\{z,y\})$.

Question 7 *Let $H = (v_1, \cdots, v_\ell)$ be a Hamiltonian closed walk of K_n, and assume there exist $1 \leq i < j < \ell$ such that $v_i = v_j$. Show that $H' = (v_1, v_2, \cdots, v_{j-2}, v_{j-1}, v_{j+1}, v_{j+2}, \cdots, v_\ell)$ is a Hamiltonian closed walk of K_n with $w(H') \leq w(H)$.*

Question 8 *Show that, if K_n admits a Hamiltonian closed walk of weight $\leq W$, then K_n admits a Hamiltonian cycle of weight at most W.*

Question 9 *Give an 2-approximation algorithm to compute an (approximate) minimum Hamiltonian cycle in K_n with an edge-weight function satisfying the triangular inequality. Explain.*

3 Minimum spanning trees vs. shortest path trees (40 minutes)

Reminder from the lecture. Recall that, given a graph G with non-negative weight on the edges, the Kruskal's algorithm considers the edges in non-decreasing order of their weight and, at each step, add the currently considered edge to the solution if it does not create a cycle. The algorithm ends when the solution is a spanning tree of G and this actually is a minimum spanning tree of G.

Let $d \geq 2$. Let us consider the following graph $G = (V, E)$ with $d+1$ nodes $V = \{v_0, v_1, \cdots, v_d\}$ such that, for any $1 \leq i \leq d$, there is an edge with weight/length d from v_0 to v_i, and, for any $1 \leq i < d$, there is an edge with weight/length i between v_i and v_{i+1}. Such a graph is depicted in Figure 1.

Question 10 *Appy the Kruskal Algorithm on G. Explain how you proceed and what is the result that you obtain.*

Question 11 *Give d different minimum spanning trees of G.*

(hint : consider the choices you had when applying the Kruskal Algorithm)

Question 12 *Let T be a spanning tree of G and let us assume that there are $1 \leq i < j \leq d$ such that $\{v_0, v_i\} \in E(T)$, $\{v_0, v_j\} \in E(T)$ and, for any $i < k < j$, $\{v_0, v_k\} \notin E(T)$.*

Show that T is not a minimum spanning tree of G.

Question 13 *Prove that there are exactly d distinct minimum spanning trees in G.*

(hint : use 12) and 13))

A spanning tree T of G rooted at some vertex r is a shortest path tree rooted in r if, for every vertex v, the path between r and v in T is a shortest path between r and v in G.

Question 14 *In this question, we assume that $d > 9$. Show that no minimum spanning tree of G is a shortest-path tree. That is, for any minimum spanning-tree T of G and for any $v \in V(G)$, T is not a shortest-path tree rooted in v.*

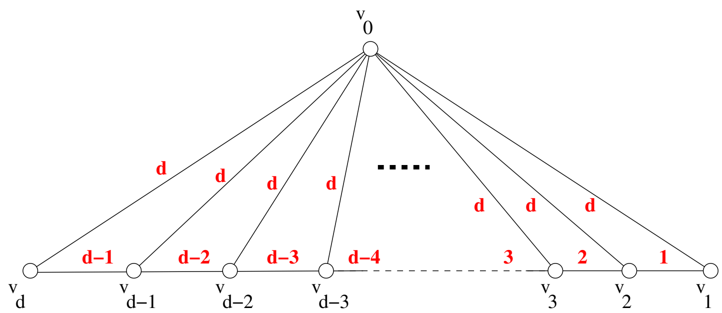

FIGURE 1 – A graph with $d + 1$ vertices ($d > 9$). A number in red indicates the weight/length of the edge it is close to.