Part 2: Multiway cut Problem

8. (1 point) Let G be a graph without multiple edges (i.e., any two vertices are linked by at most one edge) with n vertices and m edges. Show that $m = O(n^2)$.

9. (2 points) Let Π be a minimization problem and let $c > 0$. What is a c-approximation algorithm for Π? (in particular, recall the 3 properties of such an algorithm)

The goal of this part is to analyze a 2-approximation algorithm for the MULTIWAY CUT PROBLEM (defined below).

In this part, we consider a connected undirected graph $G = (V, E)$ with a weight-function on edges $w : E \to \mathbb{R}$. That is, each edge $e \in E$ is assigned a real $w(e)$, called its *weight*. Given a set $A \subseteq E$ of edges, the weight $w(A)$ of A is defined as the sum of the weights of the edges in A: $w(A) = \sum_{e \in A} w(e)$.

Cut between two vertices

Let $s, t \in V$ be two vertices of G. A subset $C \subseteq E$ of edges of G is called a *st-cut* if s and t are in two distinct connected components of $G \setminus C = (V, E \setminus C)$. That is, there are no paths from s to t once the edges of C have been removed (note that only the edges are removed, not their endpoints).

For instance, in the graph G depicted in Figure 1, the set $\{\{g, i\}, \{h, i\}\}$ is a aj-cut (that is, removing these edges disconnects vertices a and j) of weight $w(\{g, i\}) + w(\{h, i\}) = 3 + 6 = 9$.

10. (1 point) In the example of Figure 1, explain why the set $\{\{a, b\}, \{c, h\}, \{g, i\}\}$ is not a aj-cut.

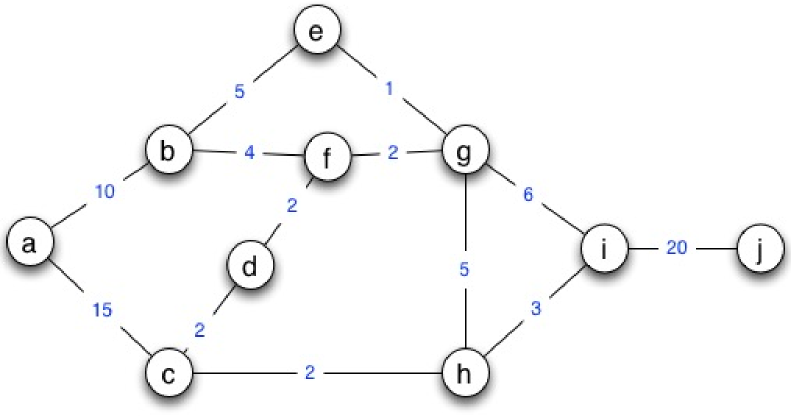

Figure 1: Example of a graph G. The integers on the edges represent their weights.

11. (1 point) In the example of Figure 1, give a aj-cut of weight at most 5.

Given a graph $G = (V, E)$ with weight w on edges and two vertices $s, t \in V$ of G, the Minimum st-cut Problem aims at computing a st-cut with minimum weight.

The Minimum st-cut Problem can be solved in polynomial-time, for instance using the Ford-Fulkerson Algorithm in time $O(m^3 \max_{e \in E} w(e))$ where $m = |E|$. **From now on, we assume that the Ford-Fulkerson algorithm can be used for solving the Minimum st-cut Problem in polynomial-time**.

Cut between one vertex and a subset of vertices

Let $s \in V$ be one vertex of G and $T \subseteq V \setminus \{s\}$ be a set of vertices not containing s. A subset $C \subseteq E$ of edges of G is called a *sT-cut* if every vertex $t \in T$ is disconnected from s in $G \setminus C$. That is, for any $t \in T$, there are no paths from s to t once the edges of C have been removed (note that only the edges are removed, not their endpoints).

12. (1 point) Let $T = \{d, h\}$. In the example of Figure 1, give a aT-cut of weight at most 9.

13. (3 points) Using the Ford-Fulkerson algorithm, give a polynomial-time algorithm that takes a graph $G = (V, E)$ with weighted edges, a vertex $s \in V$ and a subset $T \subseteq V \setminus \{s\}$ of vertices as inputs and returns a minimum weight sT-cut.

    Hint. Consider the graph G' obtained from G by adding a new vertex t adjacent to every vertex of T. Moreover, for any vertex $x \in T$, the weight of the edge $\{x, t\}$ is $w(\{xt\}) = \infty$. (see Figure 2 for an illustration of G').

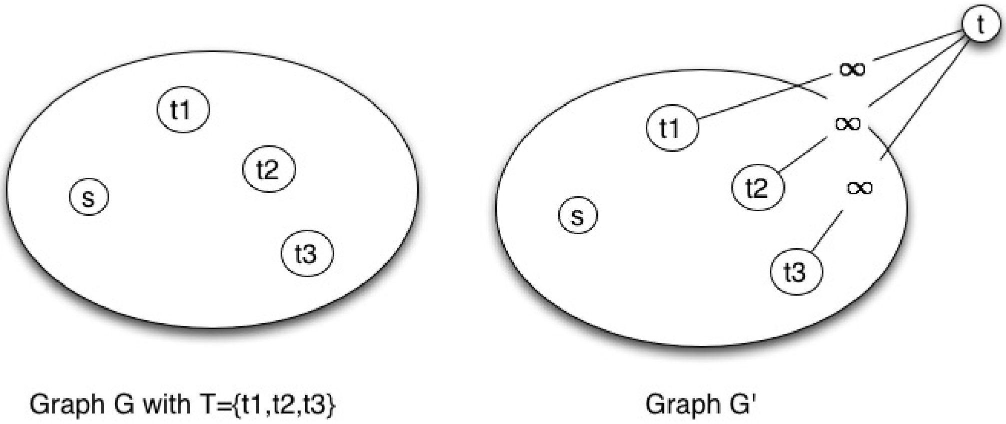

Figure 2: Example of a graph G, a vertex s, a subset $T = \{t_1, t_2, t_3\}$ and the obtained graph G'.

Multiway Cut

Let $G = (V, E)$ be a graph with edge-weight $w : E \to \mathbb{R}$, let $r \geq 2$ be an integer and $T = \{t_1, \cdots, t_r\} \subseteq V$. A T-*multiway cut* $C \subseteq E$ is a subset of edges that pairwise disconnect the vertices in T, that is, for any $i < j \leq r$, there are no paths from t_i to t_j in $G \setminus C$.

Given an edge-weighted graph $G = (V, E)$ and a subset $T \subseteq V$ of vertices, the Multiway Cut Problem consists in computing a T-multiway cut with minimum weight.

14. (1 point) Let $T = \{e, d, h\}$. In the example of Figure 1, give a T-multiway cut of weight at most 10.

Let $T = \{a, d, i, j\}$. In the example of Figure 1, give a T-multiway cut of weight at most 33.

15. (1 point) Explain why, if $|T| = r = 2$, the Multiway Cut Problem can be solved in polynomial-time.

Unfortunately, if $r > 2$, the Multiway Cut Problem becomes NP-hard.

16. (1 point) Explain what this implies (can this problem be solved in polynomial-time?).
Hint: the answer is neither yes nor no.

What follows is dedicated to a 2-approximation algorithm for this problem.

17. (3 points) Give an exponential-time algorithm that computes a minimum multiway cut in arbitrary graphs. Provide its time-complexity as a big-O of a function of m (the number of edges) and prove it.

Let us consider the following algorithm.

Algorithm 1 A 2-approximation algorithm for MULTIWAY-CUT

Require: A weighted graph $G = (V, E)$ with $w(e)$ the weight of edge $e \in E$. A set of r vertices $T = \{t_1, t_2, ..., t_r\} \subseteq V$.

for Every $i \in \{1, \cdots, r\}$ **do**
  Let $T_i = T \setminus \{t_i\}$.
  Let C_i be a minimum weight $t_i T_i$-cut in G
return $C = \bigcup_{i \leq r} C_i$.

18. (1 point) Explain why Algorithm 1 returns a T-multiway cut.

19. (1 point) Explain why Algorithm 1 performs in polynomial-time.

20. (3 points) Why Algorithm 1 is a 2-approximation algorithm for the MULTIWAY CUT PROBLEM?