CS325: Analysis of Algorithms, Fall 2016

Practice Assignment 3 Solution

Problem 1. Let MST be the minimum spanning tree of G.

(a) TRUE. Let $(A, V \setminus A)$ be a partition of V and $e = (u, v)$ be the minimum-weight edge that has exactly one endpoint in A. Suppose that $e \notin$ MST, and let P be the unique u-to-v path in MST. Let e' be an edge on P that has exactly one endpoint in A (Why does such an edge always exist?). By the definition of e we have $w(e) < w(e')$. Let $T = (\text{MST} \setminus \{e'\}) \cup \{e\}$. Then, T is a spanning tree of G that has:

$$w(T) = w(\text{MST}) + w(e) - w(e') < w(\text{MST}) \tag{1}$$

contradicting that MST is the minimum spanning tree of G.

(b) TRUE. Suppose $e = (u, v)$ is the maximum weight edge on a cycle C, and suppose, to derive a contradiction that e is in MST. By removing e from MST, we obtain two trees T_1, T_2, with vertex sets V_1 and V_2, respectively. Note that $V_1 \cap V_2 = \emptyset$ and $V_1 \cup V_2 = V$. We have $u \in V_1$ and $v \in V_2$. Therefore, there must be and edge $e' = (x, y)$ on C other than (u, v) such that $x \in V_1$ and $y \in V_2$ (Why?). By the definition of e, $w(e') < w(e)$. Let $T = (\text{MST} \setminus \{e\}) \cup \{e'\}$. Then T is a spanning tree of G that has:

$$w(T) = w(\text{MST}) + w(e') - w(e) < w(\text{MST}) \tag{2}$$

contradicting that MST is the minimum spanning tree of G.

(c) FALSE. Consider the graph in Figure 1. Red solid edges are in MST and blue dashed edges are not in MST edges. None of the edge of the blue dashed cycle is in MST.

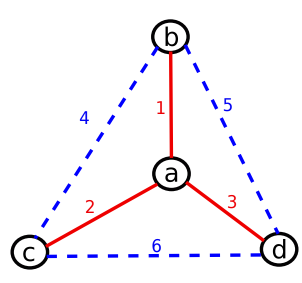

Figure 1: Red solid edges are edges of MST

Problem 2.

(a) Take a clause, say C, of the input DNF-SAT instance and assign TRUE/FALSE to variables to satisfy C. Specifically, if the literal x_i is in C, then we assign TRUE to x_i and if the literal $\overline{x}_i$ is in C, then we assign FALSE to x_i. For other variables that do not appear in C, we assign the truth assignments arbitrarily.

(b) The error lies in the fact that this reduction is not a polynomial time reduction, as the output could be exponentially long. For example, the expansion of $(x_1 \vee y_1) \wedge (x_2 \vee y_2) \wedge \ldots \wedge (x_n \vee y_n)$ to DNF-SAT has 2^n clauses.