Department of Electrical Engineering
Iris Hui-Ru Jiang

November 8, 2017

Algorithms, Fall 2017

Sample Solutions to Homework #2

1. (20) We assume the graph G is connected; otherwise we work with the connected components separately (after computing them in $O(m+n)$ time). We run BFS starting from an arbitrary node s, obtaining a BFS tree T. Now, if every edge of G appears in the BFS tree, then $G = T$, so G is a tree and contains no cycles. Otherwise, there is some edge $e = (v, w)$ that belongs to G but not to T. Consider the least common ancestor u of v and w in T; we obtain a cycle from the edge e, together with the u-v and u-w paths in T.

2. (20) Suppose that G has an edge $e = \{a, b\}$ that does not belong to T. Since T is a DFS tree, one of the two ends must be an ancestor of the other – say $a$is an ancestor of b. Since T is a BFS tree, the distance of the two nodes from u in T can differ by at most one. But if a is an ancestor of b, and the distance from u to b in T is at most one greater than the distance from u to a, then a must in fact be the direct parent of b in T. From this it follows that $\{a, b\}$ is an edge of T, contradicting our initial assumption that $\{a, b\}$ did not belong to T.

3. (15) We can sort all start and finish times in ascending order, listing start times after finish times if they are identical. Scan the list, and increment 1 when meeting a start time and decrease 1 when meeting a finish time. Through the scanning, we also record the max value and finally output the max value.

4. (25) We sort all jobs in decreasing Smith order (weighted shortest processing time w_j/t_j order, WSPT), and execute these jobs in the sorted order. The total weighted waiting time will be minimum.

   Theorem 1. *The optimal jobs schedule for the minimum weighted waiting time must be WSPT ordered.*

   Proof. Suppose there exists a schedule S is optimal and not WSPT, which means there exists two adjacent jobs j and k in S satisfying $w_j/t_j < w_k/t_k$, which implies that $w_j p_k < w_k p_j$. Consider the new schedule S' with job j and k changing order, the weighted waiting time for S' minus that of S is $w_k p_j - w_j p_k$, which is a negative number. It means that by exchanging the order of jobs j and k, we can get a better schedule S' with a smaller total weighted waiting time, which contradicts to the assumption that S is optimal. So the S must be in WSPT order. □

5. (20)

   (a) The minimum bottleneck spanning tree is not necessarily a minimum spanning tree. As shown in Figure 1. A minimum bottleneck spanning tree has total tree cost 6, while the minimum spanning tree cost is 5, which proves that a minimum bottleneck spanning tree is not necessarily a minimum spanning tree.

   (b) The minimum spanning tree must be a minimum bottleneck spanning tree. Prove as follows:
   Assume a minimum spanning tree T which is not a minimum bottleneck spanning

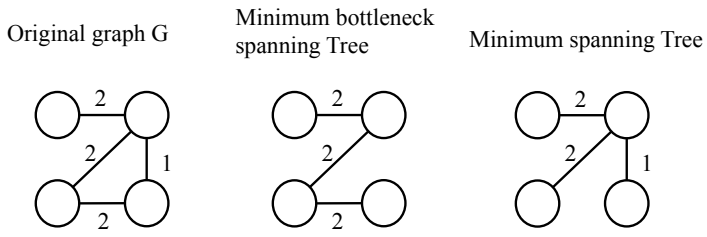

Figure 1: The counterexample of Problem 5(a).

tree T_b. The maximum weight w of edge e in T must be larger than the maximum weight w_b of edge e_b in T_b. In other words, w is larger than any edge weights in T_b. Thus, we could replace edge e by an edge in T_b and T remains a spanning tree. Then the total edge weight of T would be smaller than original T. Thus, T is impossible to be a minimum spanning tree. In conclusion, a minimum spanning tree T must be a minimum bottleneck spanning tree.