MTH 4320 Homework 7

Yaohui Wu

March 30, 2024

Problem 1

Solution. We run Prim's algorithm to find a minimum spanning tree (MST) of the graph with root d then the edges added to the MST are $\{d, e\}, \{e, g\}, \{e, h\}, \{b, e\}, \{a, b\}, \{a, c\}, \{c, f\}$ from first to last respectively. ■

Problem 2

Solution. We can use a modified Prim's algorithm to find the maximum spanning tree of G. We can find the greatest edge e instead of the lightest edge while keeping everything else in the algorithm the same. We can multiply the weight of every edge by -1 and we can use $-\infty$ instead of ∞ in the priority queue. Therefore, when we remove a vertex with the highest priority in the queue we will choose the edge with the highest weight in the original graph. The time complexity of the modified Prim's algorithm is $O(|V| \log |V| + |E|)$. ■

Problem 3

Solution. Since the weight of each edge is either 0 or 5 thus we can directly choose the lightest edge incidental to each vertex. Let u be an arbitrary vertex in G, we check all of its incidental edges. If an edge has weight 0 then we add that edge to connect the adjacent vertex v. If all the edges have weight 5 then we can add any edge. Hence, we can make sure that we select the lightest edge to connect each vertex in G. We run operations with $O(1)$ time for $|V|$ vertices and $|E|$ edges so the time complexity of the algorithm is $O(|V| + |E|)$. ■

Problem 4

Solution. We can use BFS to find e in G and this takes $O(|V| + |E|)$ time. We can compare the weight of e to the edge with the same endpoints in the MST. If the weight is lighter, then we replace the edge with e in the new MST. The time complexity of the algorithm is $O(|V| + |E|)$. ■