Exercise Set 18

1 (B)

Proving Kruskal’s algorithm relies on minimizing the maximum edge on a path aka maintaining the minimum bottleneck property (MBP). Given the information in the problem, T must satisfy the MBP as well as be acyclic. Increasing each edge cost by 1 in an acyclic graph would not affect the MBP since each edge is considered independently of the costs of other edges. Therefore, T must remain a MST since the modifications do not affect either requirement of T being a MST.

As proven in previous chapters the shortest path P may or may not remain true after the modifications to create G’. Since the shortest path considers all the edges included from s to t increasing the cost of each edge may disproportionately affect some paths more so than others.

Consider the two graphs G and G’ as shown below

A screenshot of a graph

Description automatically generated

In the original graph G the shortest path is P = S -> X -> Y -> T = 6. In our altered graph G’ P is no longer the shortest path and the new shortest path P’ = S -> T = 8

2

What is an MST. A subset of edges that connects every vertex of a graph, without any cycles, with the minimum possible total edge weight. I.E. a MST is a path that must satisfy three properties: minimum bottleneck property (MBP), acyclic, connected. Creating a path that satisfies the first two properties given a connected undirected graph (of any size) is a simple matter. Creating a path to satisfy all three properties can be done using Kruskal’s or Prim’s algorithm. So we know we can create at least one MST for every connected undirected graph G with distinct edge costs.

Next, prove that the MST for the graph is unique:

Proof by contradiction:

Consider that graph described above with two unique MST’s *A = (V, E1)* and *B = (V,E2)*. Since *A* and *B* are unique there must be some edge *ei* that belongs to one MST but not the other, in other words it belongs to the set *L = E1*­ – *E2* . Among all possible edges *ei­* in *L* choose the one with the smallest weight *e1*, let’s assume *e1* belongs to *A*. Since *B* is an MST then adding *e1* to *B* ({*e1*} U *B*) must result in a cycle *C*. Since A is an MST, it contains no cycles and *C* must have an edge *e2* not in *A* but in *B*, aka in *L*. Since *e1­* was selected when choosing the minimum edge in *L* (and all edge weights are distinct), *e1­*­ < *e2*. Thus replacing *e2* with *e1­*­ in B results in a lower edge total, contradicting that B satisfies the MBP and is thus not a MST.