CS 61BL Summer 2026

Disjoint Sets, Shortest Paths, and MSTs

Exam Prep 04: July 31, 2026

1 Union Find

(a) Assume we have nine items, represented by integers 0 through 8. All items are initially unconnected to each other. Draw the union find tree, draw its array representation after the series of `union()` and `find()` operations, and write down the result of `find()` operations using WeightedQuickUnion without path compression. Break ties by choosing the smaller integer to be the root.

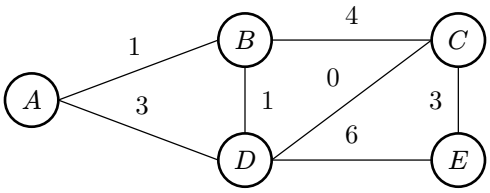
Note: find(x) returns the root of the tree for item x.

```
union(2, 3);
union(1, 2);
union(5, 7);
union(8, 4);
union(7, 2);
find(3);
union(0, 6);
union(6, 4);
union(6, 3);
find(8);
find(6);
```

(b) For each of the arrays below, write whether this could be the array representation of a weighted quick union with path compression and explain your reasoning. Assume the negative entries mean the node is root and its absolute value is the number of elements contained in the subtree.

```
     i:    0 1 2 3 4 5 6  7 8 9
   ---------------------------
A. a[i]:   1 2 3 0 1 1 1  4 4 5
B. a[i]:   9 0 0 0 0 0 9  9 9 -10
C. a[i]:   1 2 3 4 5 6 7  8 9 -10
D. a[i]: -10 0 0 0 0 1 1  1 6 2
E. a[i]: -10 0 0 0 0 1 1  1 6 8
F. a[i]:  -7 0 0 1 1 3 3 -3 7 7
```

2 Dijkstra's Algorithm



(a) Run Dijkstra's Algorithm on the graph above starting from vertex A. Break ties alphabetically.

i. Fill in how the priority values change below. When you remove a node from the priority queue, mark it with a check, and leave it blank for the subsequent rows.

ii. Sketch the resulting shortest paths tree in the end.

Node	A	B	C	D	E
Start	0	∞	∞	∞	∞
Iter 1	✓				
Iter 2					
Iter 3					
Iter 4					
Iter 5					

3 Sticky Flights

Your airline company has been contracted to fly a large shipment of honey from Honeysville to the 61Bees in Goldenhive City. However, the airplane doesn't have enough fuel capacity to fly directly to Goldenhive City so it will stop at at least one of n airports along the way to refuel. Refueling takes an hour, and if the airport is one of $k < n$ airports, your airplane will be grounded for six hours due to curfews (refueling is included in the six hours). The 61Bees want their honey as soon as possible so please design an algorithm to find the route that will allow your airplane to reach Goldenhive City in the least amount of hours.

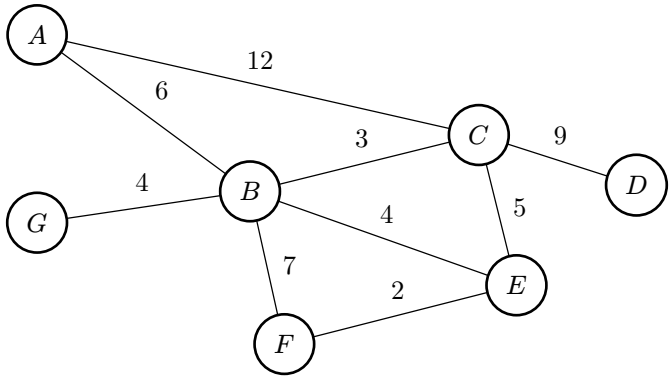
Hint: Think of the n airports as a graph, where the paths between them are edges of weight equivalent to the number of hours it takes to fly from airport A to airport B. You may assume that the amount of time it takes to fly from A to B is equal to the amount of time it takes to fly from B to A.

4 MST Algorithms

- **Prim's algorithm:** Start from any vertex. Repeatedly add the shortest edge that connects a vertex in the tree to a vertex not yet in the tree.
- **Kruskal's algorithm:** Sort all edges by weight. Add edges in order from smallest to largest, skipping any edge that would create a cycle.

(a) Fill in the blank with an expression using V (the number of vertices) and/or E (the number of edges):

Both algorithms stop when all V vertices are connected to the tree, which occurs when the tree contains _____ edges.



(b) For the graph above, list the edges in the order they're added to the MST by Kruskal's and Prim's algorithm. Assume Prim's algorithm starts at vertex A. Assume ties are broken in alphabetical order. Denote each edge as a pair of vertices (e.g. AB is the edge from A to B). The first edge is shown for you.

Prim's algorithm order: AB,

Kruskal's algorithm order: EF,

(c) True/False: Adding 1 to the smallest edge of a graph G with unique edge weights must change the total weight of its MST.

(d) True/False: If all edge weights in a graph are unique, there is only one possible MST.

(e) True/False: The shortest path from vertex u to vertex v in a graph G is the same as the shortest path from u to v using only edges in T, where T is the MST of G.

5 Multiple MSTs

Recall a graph can have multiple MSTs if there are multiple spanning trees of minimum weight.

(a) For each subpart below, select the correct option and justify your answer. If you select "never" or "always," provide a short explanation. If you select "sometimes," provide two graphs that fulfill the given properties.

1. If **some** of the edge weights are **identical**, there will

   ◯ never be multiple MSTs in G.

   ◯ sometimes be multiple MSTs in G.

   ◯ always be multiple MSTs in G.

   Justification:

2. If **all** of the edge weights are **identical**, there will

   ◯ never be multiple MSTs in G.

   ◯ sometimes be multiple MSTs in G.

   ◯ always be multiple MSTs in G.

   Justification:

(b) Suppose we have a connected, undirected graph (G) with (N) vertices and (N) edges, where all the **edge weights are identical**. Find the maximum and minimum number of MSTs in (G) and explain your reasoning.

Minimum:

Maximum:

Justification:

(c) It is possible that Prim's and Kruskal's find different MSTs on the same graph G (as an added exercise, construct a graph where this is the case!).

Given any graph G with integer edge weights, modify the edge weights of G to *ensure* that

(1) Prim's and Kruskal's will output the same results, and

(2) the output edges still form a MST correctly in the original graph.

You may not modify Prim's or Kruskal's, and you may not add or remove any nodes/edges.

Hint: Look at subpart 1 of part (a).