CSCE4133/5133 – Algorithms
Fall 2024

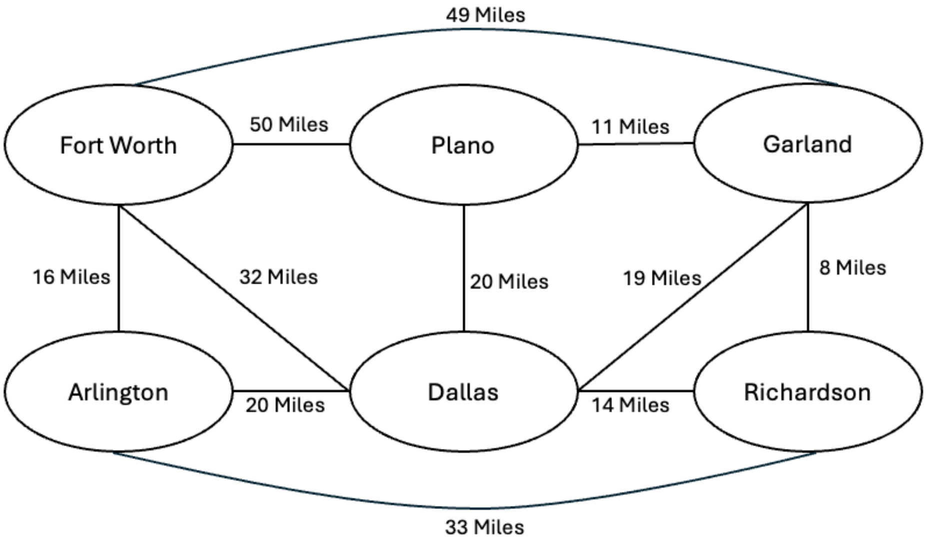
Quiz 4

Time: 25 minutes

Instructions:

- Write your full name, email address and student ID in the report.
- Submission via BlackBoard or paper

The following graph represents the travel cost between different cities.



Question 1:
Use Kruskal's algorithm to find the minimum spanning tree of the graph above (run step by step as in the class lecture).

Sort the edges:
{Garland, Richardson}, {Plano, Garland}, {Dallas, Richardson}, {Fort Worth, Arlington}, {Garland, Dallas}, {Arlington, Dallas}, {Plano, Dallas}, {Fort Worth, Dallas}, {Arlington, Richardson}, {Fort Worth, Garland}, {Ford Worth, Plano}

Note: {Arlington, Dallas}, {Plano, Dallas} have the same weight so the order can be changed.
Let X be another node.

Add the edge {Garland, Richardson}
We have connected components {Garland, Richardson}, {X}

Add the edge {Plano, Garland}
We have connected component {Garland, Richardson, Plano}, {X}

Add the edge {Dallas, Richardson}
We have connected component {Garland, Richardson, Plano, Dallas}, {X}

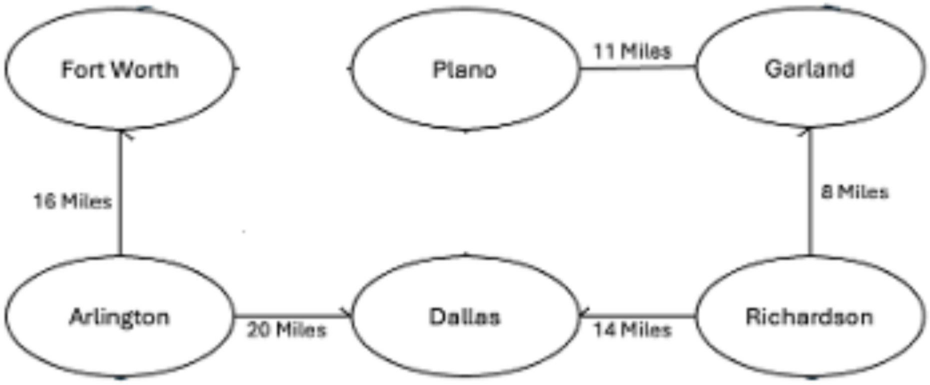
Add the edge {Fort Worth, Arlington}
We have connected components {Garland, Richardson, Plano, Dallas}, {Fort Worth, Arlington}, {X}

Since Garland and Dallas are in the same connected component, skip {Garland, Dallas}

Add the edge {Arlington, Dallas}
We have a connected component {Garland, Richardson, Plano, Dallas, Fort Worth, Arlington}

Now we can skip the other edges.

Then we have the minimum spanning tree with edges:
{Garland, Richardson}, {Plano, Garland}, {Dallas, Richardson}, {Fort Worth, Arlington}, and {Arlington, Dallas}



Question 2:
From the result in Question 1, what is the route to travel from “Fort Worth “to “Richardson” on the Minimum Spanning Tree?

Ford Worth -> Arlington -> Dallas -> Richardson

Question 3: (15 points)
Will the Kruskal’s algorithm produce a different minimum spanning tree with Prim’s algorithm? Why?

The Kruskal’s algorithm might produce a different minimum spanning tree (MST) with Prim’s algorithm. This is because the Prim’s algorithm builds the MST from a source node to another adjacent node that have the smallest weight. This makes the order of chosen edges for the MST different.

Question 4:
Given a function `int f(int n)` defined as:

```
int f(int n) {
  if (n <= 1) {
    return n;
  }
  return f(n-1) + f(n-2);
}
```

What is the run time complexity of the function f?

(A) O(n^2)
(B) O(2^n)
(C) O(n!)
(D) None of above

Question 5:
Which run time complexity is similar to O(n^3)?

(A) O($1 + 2n^2$)
(B) O($400n + 300n^3$)
(C) O($400n! + 300n^3$)
(D) O($400n + 300n^3 + 200n^5$)

Question 6:
Suppose a function `void f(int n)` has the run time complexity of O(n), what is the run time complexity of the below procedure?

```
for (int i = 1; i <= n; i++) {
  f(i);
}
```

(A) O(n)
(B) O(n!)
(C) O(2^n)
(D) O(n^2)