Arjun Kahlon

Assignment NP- Completeness

1.

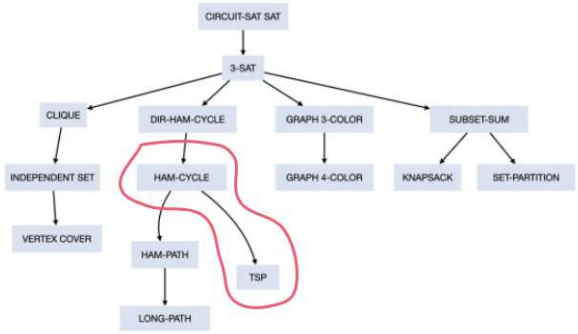
Q1

First, let's show the problem is in NP

Given a graph G, paths of TSP and the number k, we can check for all the TSP paths against graph G to verify whether all the vertices are included and the cost is no larger than k. This check can be done in polynomial time.

Second, let's show the problem is in NP-hard. Pick one known NP-hard problem and reduce it to our problem.

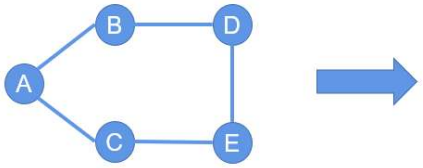
Let's pick the Hamiltonian cycle since that NP-hard problem is similar to our problem and it is easy to reduce from a similar problem to our problem.

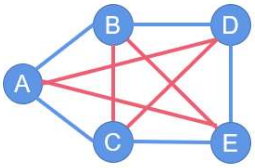


Q1- cont'

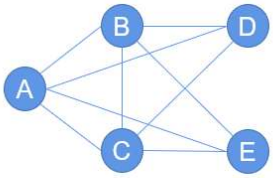
Let's take a hamiltonian cycle.

For reduction, let's turn it to a graph G with all nodes connecting others. This operation needs to do (V-1+0)*V/2-V times so it can be done in polynomial time.





reference



A's round: 4 edges
B's round: 3 edges
C's round: 2 edges
D's round: 1 edge
E's round: 0 edge

suppose we have V nodes

1st round: V-1 edges
2nd round: V-2 edges
...
...
last round: 0 edge

(v-1)+(v-2)+(v-3) +... 0 = [(v-1)+0]*v/2

Q1- cont'

Let's take a Hamiltonian cycle.

For reduction, let's turn it to a graph G' with all nodes connecting others. This operation needs to do (V-1+0)*V/2-V times so it can be done in polynomial time.

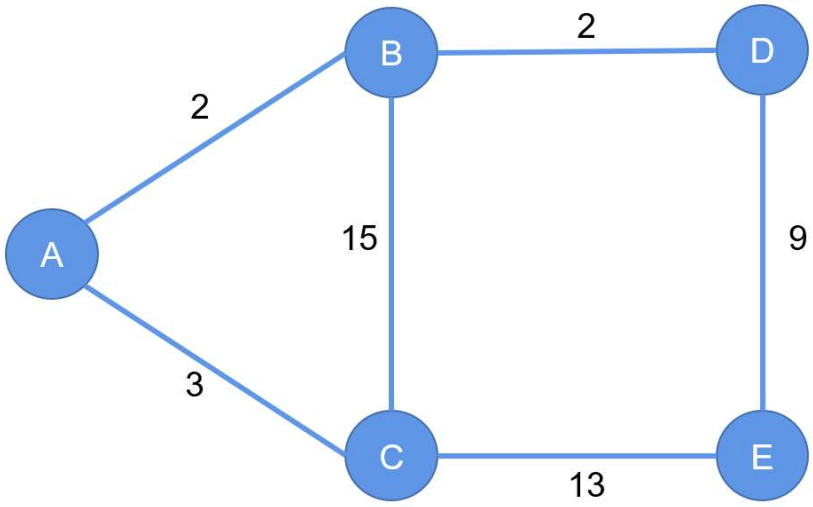
Suppose we have the 'magic polynomial time algorithm' for our TSP problem. From the graph G', we can easily remove the edges that are not included in the cost of k. By doing that, we get a Hamiltonian cycle. However we know Hamiltonian cycle is NP-Hard, so we cannot have that 'magic polynomial time algorithm'. Therefore, our problem is NP-Hard.

Now we know the problem is in both NP and NP-Hard, so it is NP-Complete.

2.

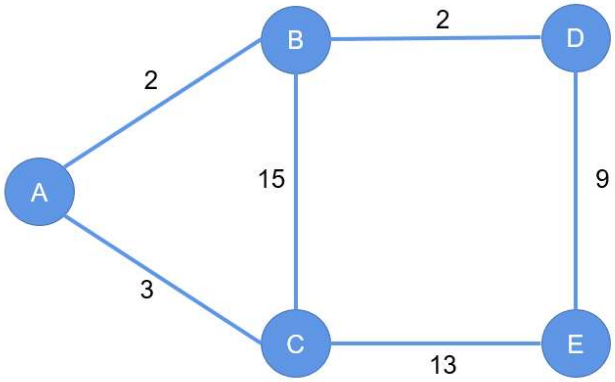
a)

Q2 - a



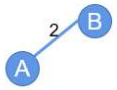
b)

Q2 - b

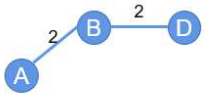


Let's use Prim's algorithm to get the MST first.

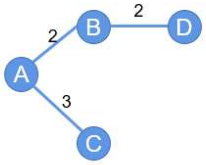
Step 1- Start from A and pick B.



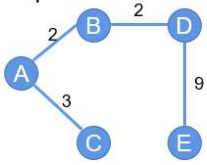
Step 2- Based on Step 1 pick D.



Step 3- Based on Step 2 pick C.

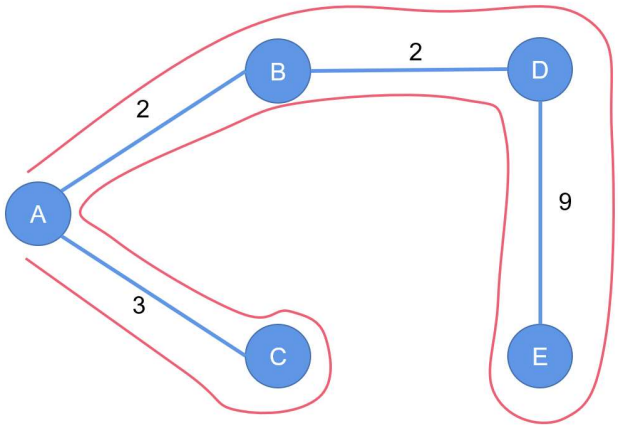


Step 4- Based on Step 3 pick E.



Q2 - b cont'

MST



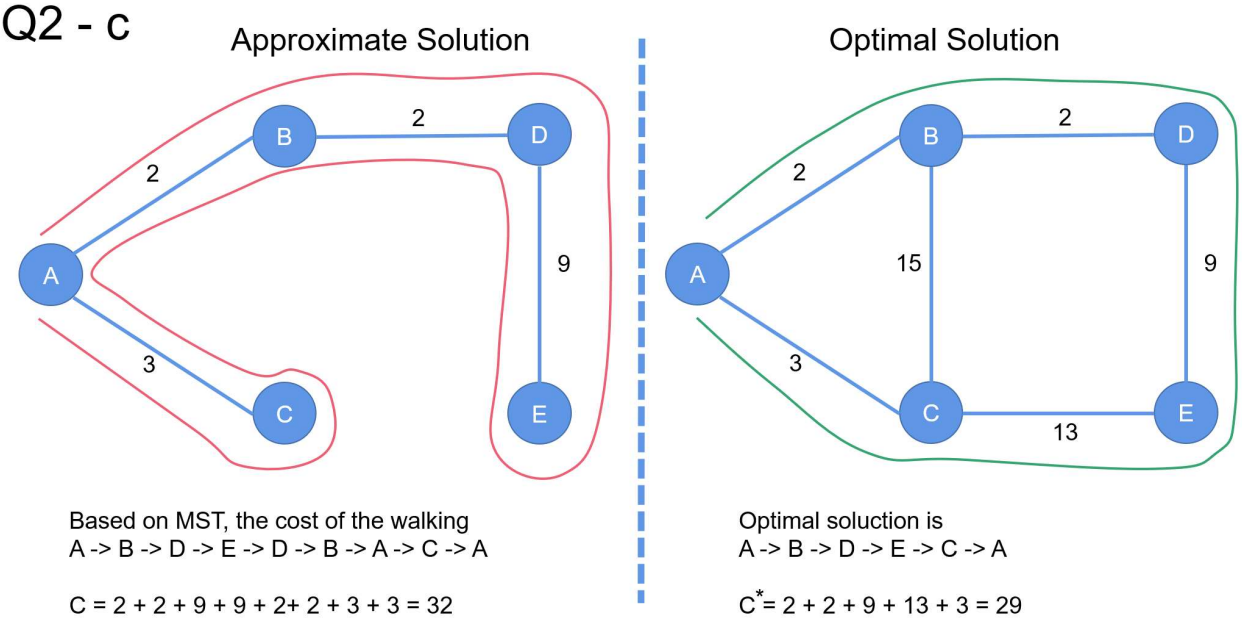
Let's walk based on the MST following below path.

A -> B -> D -> E -> D -> B -> A -> C -> A

So above is the approximate solution.

c)

Q2 - c



Based on MST, the cost of the walking
A -> B -> D -> E -> D -> B -> A -> C -> A

C = 2 + 2 + 9 + 9 + 2+ 2 + 3 + 3 = 32

Optimal soluction is
A -> B -> D -> E -> C -> A

C^{*}= 2 + 2 + 9 + 13 + 3 = 29

As a result, we get accuracy ration max(32/29, 29/32) = 1.1034 = 1.1

d)

Q2 - d

```
approx-neartest-neighbour-tsp(G):
    solution = {}
    picked_nodes = {}
    remaining_nodes = All nodes in G

    starting_point_node = randomly pick node from remaining_nodes
    picked_nodes.add starting_point_node
    remaining_nodes.remove starting_point_node

    while (remaining_nodes is not empty):
        nearest_neighbour = pick the nearest node to picked_nodes from remaining_nodes
        solution.add path to nearest_neighbour
        picked_nodes.add nearest_neighbour
        remaining_nodes.remove nearest_neighbour

    return solution
```

Debriefing

1. I found this assignment to be quite difficult and spent 15-20 hours to fully complete it.
2. Overall, I would rate it as difficult as I found the material hard to conceptually grasp. I will need to spend more time reviewing this weekend.
3. I'd say I understand about 60% of the material. I hope to gain a better understanding by the end of this week.
4. I would appreciate a webinar on this material that would supplement the lessons from the Modules.