Problem 1

Collaborators: List students that you have discussed problem 1 with: Vishnu Teja Narapareddy, Tulika Sureka, Raushan Pandey

(a) (a) Primal LP

This representation stands for finding the shortest s-t path in the given graph and the objective function tries to minimise the total cost of the path. Hence, it helps us to get the shortest s-t path in the given undirected graph.

Variable c_e denotes the capacity/weight of that edge.

Variable x_e denotes whether the edge e is present in the shortest s-t path or not.

The first constraint denotes that the number of edges taken from each cut in S' for finding the s-t path must be greater than or equal to 1, which means that the vertices s and t are atleast one edge apart.

The second constraint denotes that for each edge, it's contribution to the s-t path cannot be negative. It has to be greater than or equal to 0.

(b) Dual LP

The objective function in the dual representation here tends to maximise the contribution of each cut towards the s-t path which in turn gives us the shortest s-t path because of the constraints associated with it.

Variable c_e denotes the weight of that edge.

Variable y_s denotes the contribution of a given cut for the shortest s-t path.

The first constraint denotes that the sum of fractional contributions of each edge over all the cuts must be less than or equal to the weight of that edge.

The second constraint denotes that the contribution of every cut towards finding the s-t path cannot be negative. It has to be greater than or equal to 0.

(b) Values for the corresponding variables are:

$$
\begin{array}{lll}
C_1 = \{s\} & C_2 = \{s,b\} & C_3 = \{s,a,b\} \\
e_1 = (s,b) & e_2 = (s,a) & e_3 = (b,c) \\
y_{C_1} = 13 & y_{C_2} = 3 & y_{C_3} = 6
\end{array}
$$

(c) We can prove this by induction.

Base Case: Initially, the set F is empty as we only have root node s. So, it is true trivially.

Now, let's suppose that after k iterations, set F_k forms a tree. We need to show that after the next iteration of the algorithm, the set F_{k+1} also forms a tree.

Proof:

In the k+1 iteration, the algorithm searches for next edge which connects one of the vertices in the component C_k to one of the vertices in the set V$\setminus$ C_k. It stops at the point when we find an edge whose value is equal to y_C for that cut. Hence, this iteration adds an edge joining two disconnected components. This shows that cycle cannot be formed by the addition of this edge. Also, as the set F_k formed a tree, which means that it was connected. After addition of this edge also, the graph still remains connected. Hence, we can say that the set F_{k+1} also forms a tree.
Hence, proved.

(d) According to the question, we say that if $y_S > 0$, then $|P \cap \delta(S)| = 1$.

Now, let's suppose that it not equal to one. It is some value greater than 1. It means that the path P crosses the cut S multiple times. Consider the last such edge (u,v). As the vertex u is in the same connected component as s, so there exists a path from s to u in that component. After appending the remaining part of P from vertex v to vertex t, we would get the required s-t path. Thus, we can say that all the remaining edges in P are crossing the cut S multiple times. This means that there is cycle as we have multiple paths between s and t. However, formation of cycle is not possible, as we proved in part c above that we get a tree after every iteration in the algorithm and the path P is connected subset of the tree obtained after the last iteration in the algorithm. So, it means that P cannot have a cycle within it. This shows that it is not possible for path P to have multiple edges which cross through the same cut. This is in contraction to what we supposed initially.

Now, let's suppose that the value is less than 1. In this case, we would not be able to generate a path P as we won't be able to extend beyond that cut in the algorithm. Hence, this is not possible, as we get a path returned form the algorithm.
Hence, proved.

(e) Let say P is the path returned by the algorithm.
In part (c), we proved that the set of edges which have variable $x_e = 1$, forms a tree at the end of each iteration of the algorithm. The path P comprises of a subset of these edges, hence there is a unique path between s and t in that tree after final iteration.
In part (d), we proved that for any cut S in the set S', we say that the cut contributes just one edge to the path P.
Now, we need to prove that the path P is the shortest s-t path.
We will use the method of contradiction.
Let say, a cut S $\in$ S' which has $y_S > 0$ does not choose an optimal edge in the algorithm. Let the edge chosen is e by that cut in that iteration. According to the algorithm, we add an edge to our connected component as soon as the summation of values of all y_S become equal to the capacity for some edge in δ(S). So, this means that the iteration for the cut S would have stopped as soon as it would have found such an edge, which would be the most optimal one for the connected component C for that iteration in the algorithm. This contradicts our supposition that the cut S did not chose an optimal edge which would have led to the shortest s-t path.
Hence, proved.

Problem 2

Collaborators: List students that you have discussed problem 2 with: Vishnu Teja Narapareddy, Tulika Sureka, Raushan Pandey

(a) This problem can be solved in polynomial time.
We can use the Prim's algorithm to construct the minimum spanning tree and adding up the weights on the edges which are there in the tree. If the weight for the spanning tree comes out to be less than or equal to 42, then we can say that the graph G has a spanning tree of weight atmost 42 else we say that the graph cannot a spanning tree with weight less than or equal to 42.

TC:
The time complexity for the problem is O(ElogV), where E is the number of edges in the graph G and V is the number of vertices in the graph.

(b) We will try to show that the given problem is NP-hard problem.
We can reduce the undirected hamiltonian path problem to the given 2 leaf spanning tree problem.
Hamiltonian path is also NP-hard problem. Its reduction can be easily shown through Hamiltonian cycle. We can say that if we find a hamiltonian cycle in graph, we can get a hamiltoninan path by deleting one edge from that cycle. For converse, we can say that, if we find a hamiltonian path in the graph, then we can get a hamiltonian cycle in the graph by connecting the last vertex to the vertex from which we started the path. Hence, hamiltonian path is also a NP-hard problem.
Construction:
From the given undirected graph G = (V, E), start vertex s and end vertex t, we can construct a graph G' = (V', E') such that V' = V $\cup \{s_1, t_1\}$ and E' = E $\cup \{(s_1, s), (t, t_1)\}$. Now, we will prove that there is a hamiltonian path from s to t in G if and only if there is a spanning tree in G' with exactly 2 leaves.
This construction can be done in polynomial time.

Proof:
Let's assume there is a hamiltonian path P from s to t in graph G. As P is a spanning tree in G, we can construct P' = P $\cup \{(s_1, s), (t, t_1)\}$, which will be a spanning tree in G'. Also, every vertex in the spanning tree P in G has degree 2 except vertex s and t, which have a degree of 1. Similarly analysing P', we can say that every vertex common in graph G and G' has degree 2, including s and t, because of the edge construction done by us earlier. Only the 2 new nodes added by us s_1, t_1 have a degree of 1. Thus, we can say that the spanning tree has exactly 2 leaves.
Now, let's assume there is a spanning tree H = (V', F) of G' with exactly 2 leaves. As the new vertices s_1, t_1 added by us have degree 1 in G', they will have the same degree in subgraph H too. This means that every other node in the subgraph H must have degree greater than or equal to 2 as we assumed that H has only 2 leaves. We can say that edges in set F must also form a path from s to t because as we start from s, every node must be connected to some other node in the graph as the degree of each node is greater than or equal to 2 (shown above) and this other node cannot be in the set F as spanning trees are acyclic in nature. Hence, this gives a hamiltonian path from s to t in G.

Hence, we showed that undirected hamiltonian path can be reduced to the given problem in polynomial time. Hence, the given problem is also NP-hard.

(c) For this, we show that the given problem is equivalent to the undirected hamiltoninan path problem.
The problem in part (b) can be reduced to this problem. We can easily see that finding a spanning tree with exactly 2 leaves means that there are two nodes with degree as 1 and rest all the nodes have a degree of 2. Hence, it means that the maximum degree of every node in that spanning tree is 2. Hence, this problem is equivalent to finding a spanning tree with exactly 2 leaves, which we already showed in the previous part, that it is NP-hard problem.

2-Spanning tree means a spanning tree in which each vertex has degree less than or equal to 2 in the path. This statement is equivalent to finding undirected hamiltonian path in the graph G as all the vertices in the path should also have a degree atmost 2.
Hence, we can say that the given problem is NP-hard.

(d) We can prove this problem as a NP-hard problem.
We can reduce undirected Hamiltonian path problem to the given problem, which we already know that it is a NP-hard problem.

Construction:
Construct a graph G' from G in the following manner:

- We add a vertex z with edges to every other vertex in graph G.
- Now add 41 vertices,namely, l_1, l_2, ..., l_{41}, each with edges to z.

It can be clearly seen that G' can be built from G in polynomial time.
Let say, we refer to a spanning tree of G' as spanning-42 if it has atmost 42 leaves. We can say that G contains a Hamiltonian path if and only if G' contains an approx-hamiltonian spanning tree.
Proof:
Let G has a hamiltoninan path P which starts at vertex s and ends at vertex t. Let T be the subgraph of G' obtained by adding the edge tz and all possible edges zl_i. Then we can say that T is a spanning tree of G' with exactly 42 leaves, which are s and the newly added 41 vertices l_i.
Suppose G' has spanning-42 spanning tree T. This means that the spanning tree T has atmost 42 leaves. Each node l_i is a leaf in T, so T must consist of 41 edges zl_i and a simple path from z to some vertex s in G. Let t be the only neighbour of z in the subgraph T that is not in leaf l_i, and let P be the unique path in the subgraph T from s to t. This path visits every vertex of G, which means that P is a Hamiltonian path in G.
Hence, we showed that undirected hamiltonian path can be reduced to the given problem in polynomial time.
Hence, the given problem is also NP-hard.