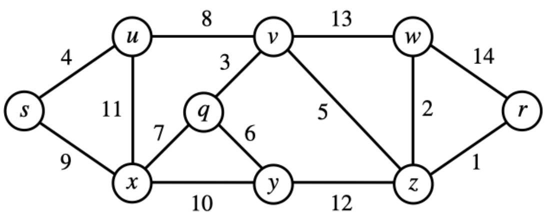
Advanced Data Structures and Algorithms
Quiz 3

For each of the below statements answer whether it is True, False, or Unknown (if statement's truthness is not known). For each statement, you will get **2 mark** for the right answer and **-1** for the wrong answer. Write the answers on this sheet and return it back. You do not need to submit the rough sheets. Make sure to read through all the questions, as they are **NOT** arranged in ascending order of difficulty.

Name: **Roll No:**
Duration: 30 Minutes **Marks:**

1. If we run Kruskal's algorithm on the below graph, then computed MST will contain the edges $\{\{q,x\},\{z,r\},\{s,u\},\{q,y\},\{v,z\},\{z,w\},\{u,v\},\{q,v\}\}$.



True.

2. The below algorithm computes the maximum spanning tree, that is, spanning tree with largest weight in A.

```
MaxST(G):
for each vertex v ∈ V(G)
    Make-Set(v)
A = ∅
sort the list of edges into monotonically decreasing order by their weight
for each edge {u, v} taken from the sorted list in order
    if Find-Set(u) ≠ Find-Set(v)
        A = A ∪ {{u, v}}
        Union(u, v)
return A
```

True: The proof of correctness for minimum spanning tree will work for maximum spanning tree as well.

3. Prim's algorithm may not give the right answer when graph contains negative edges.
False: Prim's and Kruskal's algorithm work for negative weight edges as well.

4. Let $\mathit{SPATH} = \{\langle G, s, t, k\rangle \mid G$ contains a path of length at most k from s to $t\}$. Then $\mathit{SPATH} \leq_p \mathit{SAT}$.
True: SPATH is solvable in P and hence in NP as well. Since SAT is NP-complete SPATH can be reduced to SAT in polytime.

5. A dominating set for a graph G is a subset D of its vertices, such that any vertex of G is in D, or has a neighbour in D. Then, for any graph G, every vertex cover of G is also a dominating set of G.
False: Consider a graph of just two vertices with no edge. Vertex cover is null set, but dominating set must contain both the vertices. However, the statement is true when graph is connected.

6. There exist graphs on which **Approx-VC** will always give suboptimal vertex cover.
True: Consider any graph that has a vertex cover of size 1. Approx-VC always returns vertex cover of size at least 2.

7. Let L be any decision problem. If $L \leq_p \mathit{SAT}$, then $L \in \mathsf{NP}$.
True: Because SAT is in NP. After converting an instance of L to SAT, we can use the verifier of SAT to show that L is also in NP.

8. $\mathsf{NP} \subset \mathsf{P}$.
False: Because the statement is saying that there are problems in P which are not in NP. But we know that every problem in P is also in NP.

9. The set of edges picked in **Approx-VC (**whose ending vertices were added to vertex cover) forms a maximum matching in the input graph.
False: Consider a three length path as a graph. **Approx-VC** may pick the middle edge only and terminate. But maximum matching contains two edges: first and third.

10. $\mathit{MetricTSP} = \{\langle G, c, k\rangle \mid G$ is an undirected complete graph, $c : V \times V \to \mathbb{N}$ is a cost function on the edges such that it satisfies the triangle inequality, and $k \in \mathbb{N}$, such that G has a travelling salesperson tour of cost at most $k\}$ is **NP-complete**.
True: Use the same reduction as we have seen from Hamcycle to TSP. But instead of giving 0 cost to original edges, give cost 1, and instead of giving 1 cost to new edges, give cost 2.