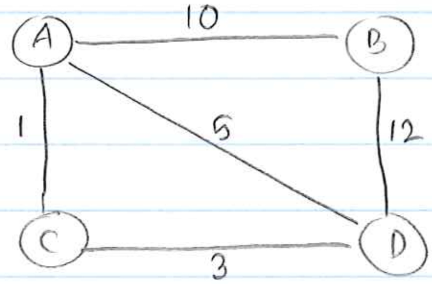
1. <u>proof</u>: Suppose our connected undirected graph G, with edges each of unique weights, has two distinct and acceptable minimum spanning tree T and T'.
And let $t = W(T) = W(T')$
Next consider the overlap of T and T'. In this overlap we'll see some cycle c with k edges, where $k-1$ edges are from T and $k-1$ edges from T'. Consider the heaviest edge e on this cycle, and assume without loss of generality that e is definitely in T. Because all edge weights are unique, for any given cycle from our original graph, an MST will not use the heaviest-weighted edge on this cycle. Hence e doesn't belong to any MST, contradicting T is an MST.

2.a) YES.
<u>proof</u>: Let T be an MST for G and T is not optimal solution for the second objective function.
Assuming T is not optimal for the second objective function there must be an edge e that is more costly than the minimum weight possible between A and $V \setminus A$, two components connected by edge e.
So we must have some other lowest cost edge e' between A and $V \setminus A$ to ensure the optimal solution for second objective function.
Let us remove e from T and add e' to get T'.
But T' is a spanning tree and has total weight less than T, contradicting T is an MST.

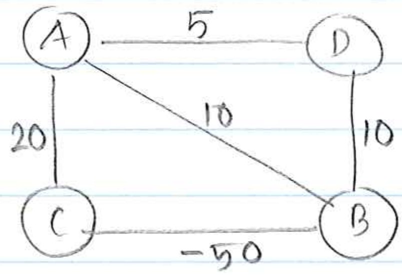
2.b) NO.

counter example:



The solution for the second objective function:
(A, B), (A, D), (A, C) with total cost 16.
But the solution for the second objective function is
(A, D), (A, C), (A, B) with total cost 14.

3.a)



For the above graph dijkstra will produce incorrect output while computing shortest path from source A through following steps;

1. update d[D] = 5, d[B] = 10, d[C] = 20
2. mark D as visited with no update
3. mark B as visited by updating d[C] = -40
4. mark C as visited with no update.

Thus it produces wrong result for B and D.

3.b) Dijkstra's algorithm always choose the unvisited node with smallest key and only updates the node that is adjacent and unvisited. As it follows this greedy approach it is not able to recalculate decreased past cost for an already visited marked node. And this decreased path cost can happen in the presence of a negative cycle as well as in mere presence of a negative edge.

3. c) For the correctness of dijkstra in case of negative weight edges from source vertex it is sufficient to show that $d[v] = \delta(s, v)$ for every $v \in V$ when v is added to S Given the shortest s~v path and given that vertex u precedes v on that path, we need to verify that u is in S. If $u = s$ then certainly u is in S. For all other vertices we have defined v to be the vertex not in S that is closest to s. Since $d[v] = d[u] + \omega[u, v]$ and $\omega(u, v) > 0$ for all edged except possibly those leaving the source, u must be in S since it is closer to s than v. There is always a simple shortest path from s to u unless their is a negative weight cycle.