1 A

The problem resembles the Minimum Spanning Tree problem, with the addition of having to calculate the complete weight of the inverse edges to the edges used in the spanning tree, and finding the largest of those two values. This value has to be smaller than some given number B.

The given problem is a graph with 3 nodes and three edges, so all nodes are connected to each other. The problem will return true, since the spanning tree with edges e_1 and e_3 have a complete weight of 4, and the reverse edges, which are also e_3 and e_1 will have a complete weight of 4 also. If any other spanning tree is chosen, either the spanning trees complete weight or the complete weight of the inverse edges will be 5 or more.

2 B

2.1 Algorithm

- Let the string R consist of edges in G: R = r_1,r_2,...,r_l.
- If the number of edges in R does not equal n-1, where n is the number of vertices in the input graph G, then return NO.
- Check whether the edges in R form a spanning tree in G. If not, then return NO.
- Let the string Q consist of the inverse edges of the edges in R, such that if r_1=e_k, then q_1=e_{m+1-k}: Q = q_1,q_2,...,q_l.
- Calculate the complete weight of the spanning tree formed by the edges in R.
- Calculate the complete weight of all the edges in Q.
- If the complete weight of all the edges in R is smaller than B, and the complete weight of all the edges in Q is smaller than B, then return YES. Else return NO.

2.2 Conditions

Assume answer is YES

- Then there exists a spanning tree made up of edges in G, with the complete weight of the spanning tree being less than B, and the complete weight of the inverse edges to the edges in the spanning tree also being less than B.

- Construct a string of edges $R^* = r_1, r_2, ..., r_l$ containing all the edges in the spanning tree
- When the algorithm receives R^*, it will construct the spanning tree, calculate the weight of it, and calculate the weight of the inverse edges and return YES.
- Therefore there is a string of length n-1 that will return YES. The probability of creating it is positive.

Assume the answer is NO

- Then no set of edges can create a spanning tree where both the complete weight of the spanning and the complete weight of the inverse edges of the edges in the spanning tree will be less than B.
- If the length of R is not n-1 then it will return NO.
- If the length of R is n-1, then the algorithm will check to see whether the edges in R form a spanning tree. If not then it will return NO.
- If they do form a spanning tree, then it will calculate the complete weight of the spanning tree and the complete weight of the inverse edges.
- These values are compared to B
- Since both weights cannot be less than B, it will return NO.

2.3 Running time

- It is checked whether there are n-1 edges in R. Time: O(n).
- It is checked whether the edges in R form a spanning tree in G. Time: O(n).
- Q is created from the inverse edges of the edges in R. Time: O(n).
- The complete weight of R is calculated: Time O(n).
- The complete weight of Q is calculated: Time O(n).
- The complete running time is therefore O(n).