b. time complexity: O(E log E) + O(E log n)

because:

sorting the edges in decreasing order : O(E log E),

going through the edges with a loop: O(E)

each find and union is O (log n)

space complexity : O(n) because we use extra data structure to keep track of parent, height and adjacency list which they need spaces based on number of nodes

C.

In Kruskal algorithm : time complexity : O(E log E + E log n) and space complexity : O(n)

In prim algorithm : time complexity: O (E log n) and space complexity is: O(n + E)

In reverse Kruskal algorithm: time complexity : O(E log E) + O(E log n) and space complexity : O(n)

Prim has a better time complexity than two other algorithms when the graph is sparse but when the graph is dense prim has a worse time complexity than two other algorithm.

Choosing the appropriate algorithm depends on the situation, data we have and our priorities.