#3

Assign arriving box with bi (i=1 to n), each box has a weight wi (i=1 to n);

Consider k is the number of tracks that b1 – bi can be fited in to; The algorithm is to put all the box in tracks as much as possible, other algorithm is to put part of box in to like 1st track, leave some put in latter box, leave 1st track not full.

When k = 1, the greedy algorithm is obviously better them other cause it only need 1 track.

Now assuming it holds for k-1; then the greedy algorithm can put N1 box into k-1 tracks, but others can only put N2 (N2<N1) tracks. For Kth track, other algorithm will put b(N2+1) b(N2+2) … b(N2+t) these t boxes. However, the greedy algorithm can put b(N1+1) … b(N1+t) cause N1>N2, so still N1+t > N2+t, which mean the it holds for k. This completes the induction.

#4

t=1;

for i= 1 to n

if s[i] == s’[t]:

t++;

endif

endfor

if t==m

return true;// s’ is subsequence of s

else

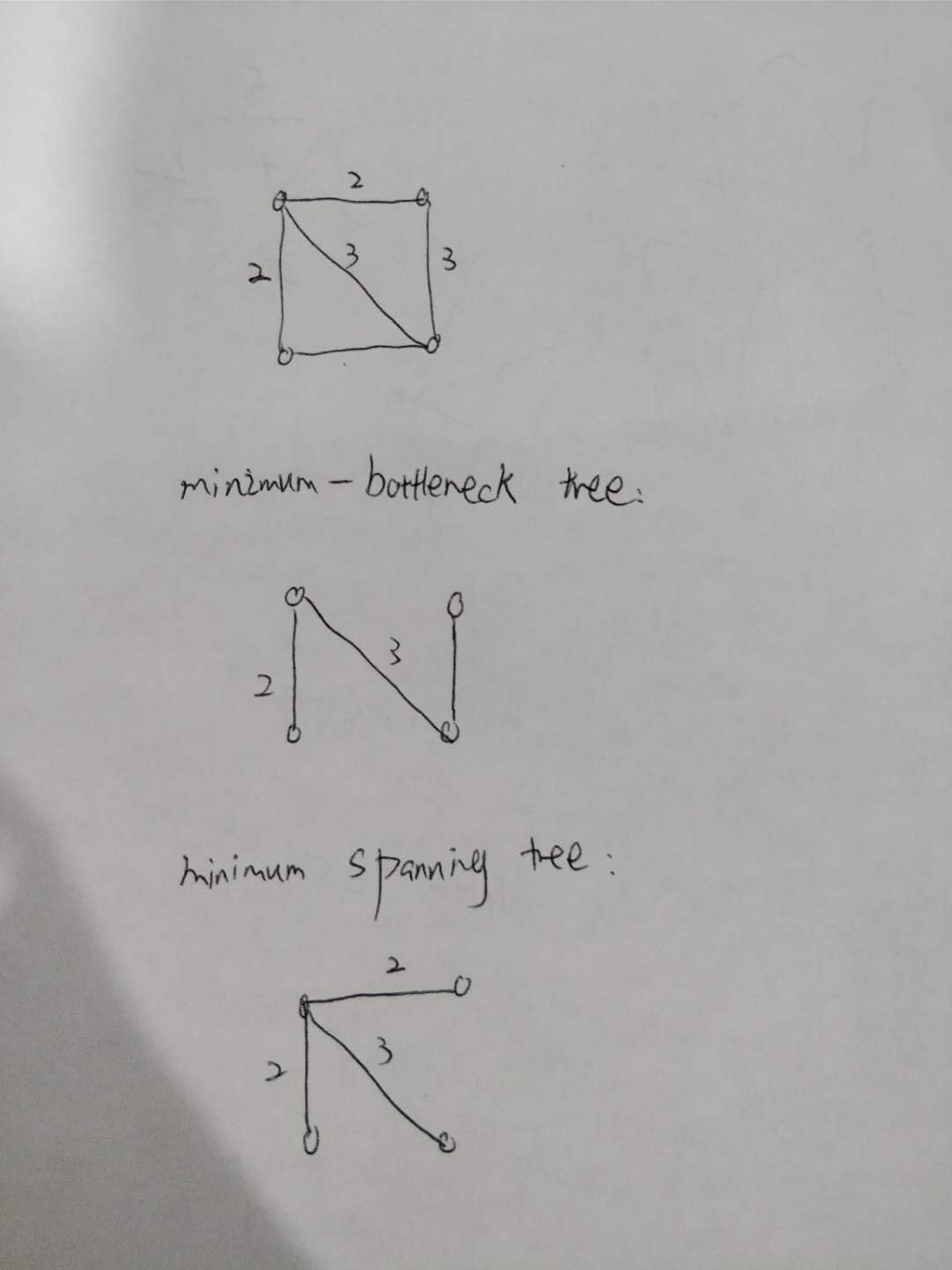
return false;

The running time of algorithm above is O(n).

#8

Using Kruskal, we add edges ascendingly. If there are two different spanning tree, in some step of Kruskal there should be have two edges with same costs can be added, but all edges’ cost are distinct, so G only has one spanning tree.

#9



1. False; counterexample is above;
2. Suppose T is minimum spanning tree but T’ is minimum bottleneck tree instead. T must have an edge that costs more than every edges in T’; put e in T’, then it forms a cycle in T’ that e is the heaviest edge in cycle. By cut property, e cannot be in T which contradict to assumption. T is minimum bottleneck tree.

#11

Using e1…em to denote edges and e(m-n+1),…em belong to T. if there are some edges in e1 to e(m-n) that equals to edges in e(m-n+1)…em then list them after the edge in the latter list, then we can get the valid ordering of edges and we can use it to generate valid execution.

#17

Using I1 … In to denote the n intervals. For interval Ij, Ij restricted solution will include Ij. For fixed j, let x be a point in Ij, delete Ij and all the intervals that overlap it we can compute Ij restricted. Then, we compute all Ij restricted for every j then pick the largest of them. And the running time of algorithm is O(n^2).

#27

Using T=(V,E) and T’=(V’,E’) to denote two spanning trees of G. |E-E’|=|E’-E|=k.

When k = 1, it’s true. When |E-E’|=k>1, e’ belongs to E’-E, adding e’ in T then we get T’’ will contains a cycle C, and the cycle contain an edge e which is not belongs to E’. Delete e in T’’ we get T’’’ and |E’’’-E’|=|E’-E’’’|=k-1. So, there is a path of length k-1 from T’’’ to T’. Since T and T’’’ are neighbors, so there is a path of length k from T to T’.