CS 260 Fundamentals of the Design and Analysis of Algorithms **Fall 2016**

HomeWork 4

Name: *Liangjian Chen*
ID: *#52006933* *October 24, 2016*

Problem 3 **Solution:**

let's assume that array A is the array of all package's weight stored by their coming order. Function $MinTruck(X)$ is the minimum number of truck to send all package in A. $A[a:]$ means the A's sub-array which its index from a to the end.

First I wan to argue that $MinTruck(A[a:]) \leq MinTruck(A[b:])$ if $b \leq a$.Consider the best arrangement of $MinTruck(A[b:])$. Taking all items in $A[a:b]$ out and after that if a truck is empty then discard it. Now we get an possible arrangement of $A[a:]$ which called X. So, $MinTruck(A[a:]) \leq X \leq MinTruck(A[b:])$ holds.

Then, let's said after arrange some trunks, company still have $A[b:]$ item left. If company make last truck less full, which means it push some item before b, and let's say it become $A[a:]$. As I argue before $MinTruck(A[a:]) \leq MinTruck(A[b:])$ always holds. Thus company can never make it better in this way.

Problem 4 **Solution:**

```
def Subsequence(S, S_prime):
  j = 0
  for event in S:
    if S_prime[j] == event:
      j += 1
    if j == len[S_prime]:
      break
  if j == len[S_prime]:
    return "Yes"
  return "No"
```

Problem 8 **Solution:**

Assume that T and T' is two different minimum spanning trees. An edge $e \in T$,and $e \notin T'$. Let's add e into Tree T, then there are two different situation.

(a) e is the most expensive among the circle
Because all weights are different, at least one edge's weight larger then e, let's assume it is e'. Substitute e for e', it forms a new tree with smaller weight. Contradict with T' is a minimum spanning tree.

(b) e is not the most expensive among the circle
There must be an edge in the circle which is not belong to tree T(Otherwise e form a circle in the tree T).Let assume it is e''. By substituting e'' for e in tree T, we construct a new tree with smaller sum weight. It contradicts with the assumption.

In conclusion, G's minimum spanning tree is unique.

Problem 9 **Solution:**

(a) No, like a graph with 3 nodes (a,b,c) and 3 edges ($e(a,b,0), e(a,b,1), e(b,c,100)$). $e(a,b,1), e(b,c,100)$ form a bottleneck tree, but obviously it is not a minimum spanning tree.

(b) Yes. Assume that T is a minimum spanning tree and T' is a cheaper bottleneck edge. Then picking out the most expensive edge, this edge is heavier than all the edges in T'. However, when we insert it into T', it must form a circle. Since this edge is the most expensive edge in this circle, it does not belong to any minimum spanning tree, which contradict with the assumption.

Problem 11 **Solution:**

Yes. If we subtract every edges in tree T a random small number δ which does not change the order of non-equal-weight edges. Then run the Kruskal algorithm, algorithm would maximum the subtraction, and return the minimum panning tree T.

Problem 17 **Solution:**

Dynamic programming could be used to solve this problem. Define state $f[i]$ indicate that begin to the very beginning to time point i, the maximum event you can have. Then transfer function is:

(a) if no event end in i: $f[i] = f[i-1]$

(b) if event x end in i:
Assume $B[x]$ is the beginning time of event x.
$f[i] = \max\{f[i-1], f[B[x]-1]+1\}$

Since all events' end time is distinct, only one event would end in a specific time point. Then, since time is a continuous variable, we can not handle infinite state space. But we can see that only the beginning time and ending time matter in this transfer equation. So now we can limit state space to a set which only contain the beginning and end time of all events. Thus, state space is $O(n)$, for each state, transfer function costs $O(1)$, the over all time complexity is $O(n)$.

Problem 27 **Solution:**

Yes, $\mathcal{H}$ is connected.
Suppose, T is original spanning tree, and T' is the target spanning tree.
$E_T = \{e | e \in T, e \notin T'\}$.
$E_{T'} = \{e | e \in T', e \notin T\}$.
First, I want to prove that it is always possible that delete one edge in E_T and add an edge in $E_{T'}$ to form a new spanning tree.

Let's choose a edge $e' \in E_{T'}$, and add it to T and form a circle. Beside e',there must be at least one edge in this circle is in set E_T. Otherwise, there would be a circle in tree T'. So this transfer operation is always doable.

Then, it is obvious that $|E_T| = |E_{T'}|$, so if we keep pick a edge in $E_{T'}$ and add it to T, eventually E_T and $E_{T'}$ would be empty which indicate tree T is transfered to T'. And a path between two arbitrary spanning tree T and T' could be found which indicate that $\mathcal{H}$ is connected graph.