Solution to HW 9

April 23, 2013

1. Let $t[i]$ be the be time taken for job $j[i]$.
   Divide the customer importance with time taken and then sort them in decreasing order. Let $r[i]$ be the sorted order of jobs in increasing order

```
Min_Schedule(job*1...n+)
{
sum = 0;
//Calculate the ratio
for(i = 1 to job.size)
r[i] = w[i]/ t[i];
// Sort the jobs based on the ratio r in decreasing order
HeapSort(job[])
f[1] = t[1];
sum = w[1]*f[1];
//Calculate the sum based on the ratio
for(i = 2 to job.size)
f[i] = f[i-1] + t[i];
sum = sum + w[i]*f[i];
return sum;
}
```

   Time complexity is the same as that of sorting which is $O(nlogn)$.

2.

```
SubSequencePresent(s[],A[])
{
found = true; //Initialize found to true i.e. sub sequence s[] in found A[]
pos = -1
// Index for traversing through string A[]
for (i 1 to s.size && found == true)
{
for (j= pos+1 to A.size)
{
if(s[i] == A[j])
```

```
found = true;
pos = j;
break;
else
found = false;
}
}
return found;
}
```

The first for loop iterates all the elements of string s. Second for loop starts from the last found character of $s[]$ in $A[]$ until reaches the end of $A[]$. So string $A[]$ is also traversed only once. Therefore, total time complexity $= O(|s| + |A|)$

3. ```
ChangeCoin(V,d[])
{
Declare an array change[] of size equal to the number of different denominations i.e. 4
for (i =1 to change.size)
change[i] = 0; // Initialize the change array to zero
for (i=1 to d.size)
{
if (V > 0)
{
change[i] = V / d[i];
V = V%d[i];
}
else
break;
}
return change[];
}
```

4. (a) We how that MST is unique when all edges have distinct weight by contradiction:

   Let us assume a Minimum Spanning Tree say $T$ is not unique. Then there is another spanning tree $S$ of equal weight as $T$. Let an edge $e$ be in Tree $T$ but not in $S$. Adding the edge $e$ in MST $S$ will make a cycle $C$. Cycle $C$ in $S$ will contain at least one edge $e'$ which will not be in MST $T$. Let us assume that the weight of edge $e$ is less than edge $e'$. So we can replace $e'$ with $e$ in MST $S$ now giving an another tree say $S'$ with smaller weight. Above contradicts our assumption that $S$ is a MST.

   (b) In the prrof given in the previous example assume that $e$ and $e'$ have the same cost and that they are the minimum weighted edges, so now
```

we have 2 MSTs T and S both with the same weight now depending on the ordering of the algorithm we get either T or S.