# LAB 2: Greedy Method

## Date : 21.2.18

## Submitted By : Abhineet Singh

1. ***Write a program that implement a fractional Knapsack problem and find an optimal solution to the knapsack Problem with instance n=7, m=15, (p1, p2,.....,P7) = (10, 5, 15, 7, 6, 18, 3), and (w1, w2,.....,w7) = (2,3,5,7,1,4,1).***

```
#include <bits/stdc++.h>
using namespace std;

struct Article
{
   int p, w;
} ;

bool cmp(struct Article a, struct Article b)
{
   double r1 = (double)a.p / a.w;
   double r2 = (double)b.p / b.w;
   return r1 > r2;
}

double Knapsack(int W, struct Article arr[], int n)
{
   sort(arr, arr + n, cmp);

   int curWeight = 0;
   double finalvalue = 0.0;
   double x[n]={0};

   for (int i = 0; i < n; i++)
   {
      if (curWeight + arr[i].w <= W)
      {
         curWeight += arr[i].w;
         finalvalue += arr[i].p;
         x[i]=1;
```

```
        }
       else
        {
            int remain = W - curWeight;
            finalvalue += arr[i].p * ((double) remain / arr[i].w);
            x[i]=(double) remain / arr[i].w;
            break;
        }
    }

 for(int i=0;i<n;i++)
        cout<<x[i]<<endl;
        cout<<endl;

    return finalvalue;
}

int main()
{
    int W = 15,n=7;
    Article arr[n] = {{10,2}, {5,3}, {15,5}, {7,7}, {6,1},{18,4},{3,1}};

    cout << "Maximum value we can obtain is "<<endl<<Knapsack(W, arr, n);
    return 0;
}
```

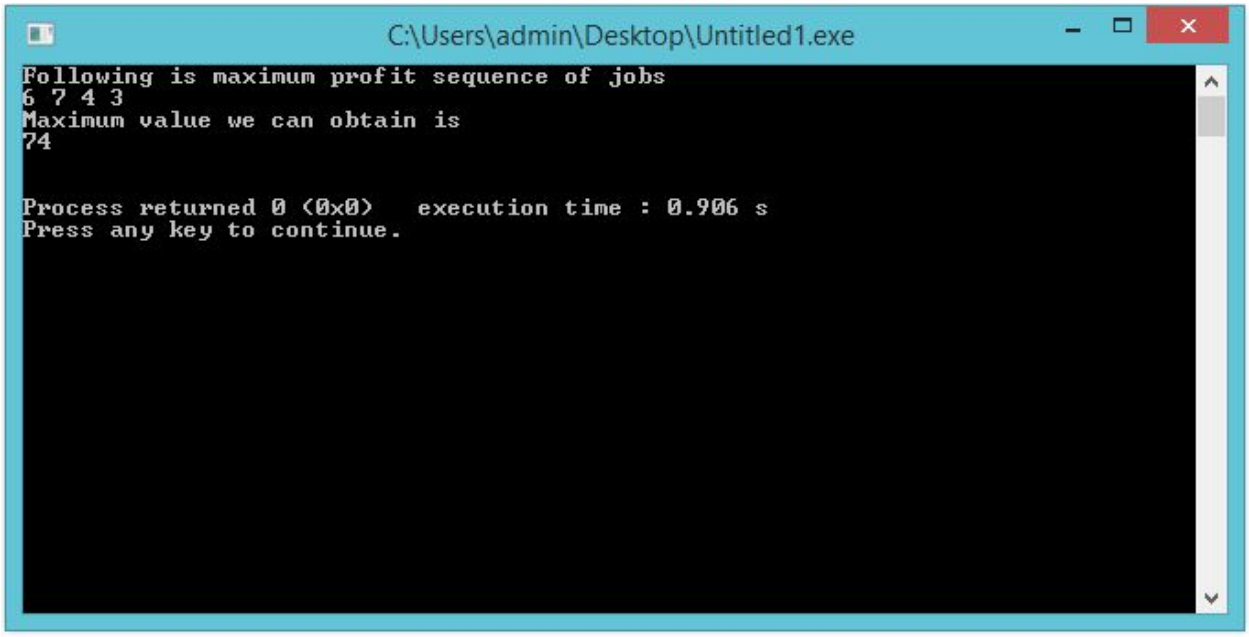
C:\Users\admin\Desktop\Untitled2.exe

```
1
1
1
1
1
0.666667
0

Maximum value we can obtain is
55.3333
Process returned 0 (0x0)   execution time : 2.452 s
Press any key to continue.
```

2. ***Write a program that implement a job sequencing with deadline problem and find an optimal solution of sequence of the job to be executed with instance n = 7, (p1, p2,.....,P7) = (3,5, 20, 18, 1, 6, 30), and (d1, d2,.....,d7) = (1, 3, 4, 3, 2, 1, 2).***

```
#include<iostream>
#include<algorithm>
using namespace std;

struct Job
{
  int id;
  int dl;
  int p;
};

bool comparison(Job a, Job b)
{
    return (a.p > b.p);
}

void job(Job arr[], int n)
{

   sort(arr, arr+n, comparison);

   int result[n];
   bool s[n];

   for (int i=0; i<n; i++)
       s[i] = false;

   for (int i=0; i<n; i++)
   {

      for (int j=min(n, arr[i].dl)-1; j>=0; j--)
      {

         if (s[j]==false)
         {
            result[j] = i;
            s[j] = true;
```

```
            break;
          }
        }
    }

   int pro=0;
    for (int i=0; i<n; i++){
       if (s[i]){
         cout << arr[result[i]].id << " ";
    pro=pro+arr[result[i]].p;
}
}
cout<<"\n"<<pro;
}

int main()
{

    Job arr[] = { {1, 1, 3},{2, 3, 5},{3, 4, 20},{4, 3, 18},{5, 2, 1},{6, 1, 6},{7, 2, 30}};
    int n = sizeof(arr)/sizeof(arr[0]);
    cout << "Following is maximum profit sequence of jobs :  ";
    job(arr, n);
    return 0;
}
```


C:\Users\admin\Desktop\Untitled1.exe
Following is maximum profit sequence of jobs
6 7 4 3
Maximum value we can obtain is
74
Process returned 0 (0x0)  execution time : 0.906 s
Press any key to continue.