Govt. Women Engg. College, Ajmer
Mid Term Examination I- Mar.8, 2018
Design and Analysis of Algorithm

Max. Marks:20 Time:1Hr.

Q.1 Solve 0/1 knapsack problem giving the recursive formula of dynamic programming for following table of items with knapsack capacity of 50 kg. Can you solve it using greedy approach also? Give reason in short. [4+4= 8 marks]

Item (sealed packets of daal)	"Urad"	"Masoor"	"Chana"
Weight (kg)	10	20	30
Price(Rs)	800	1200	2100

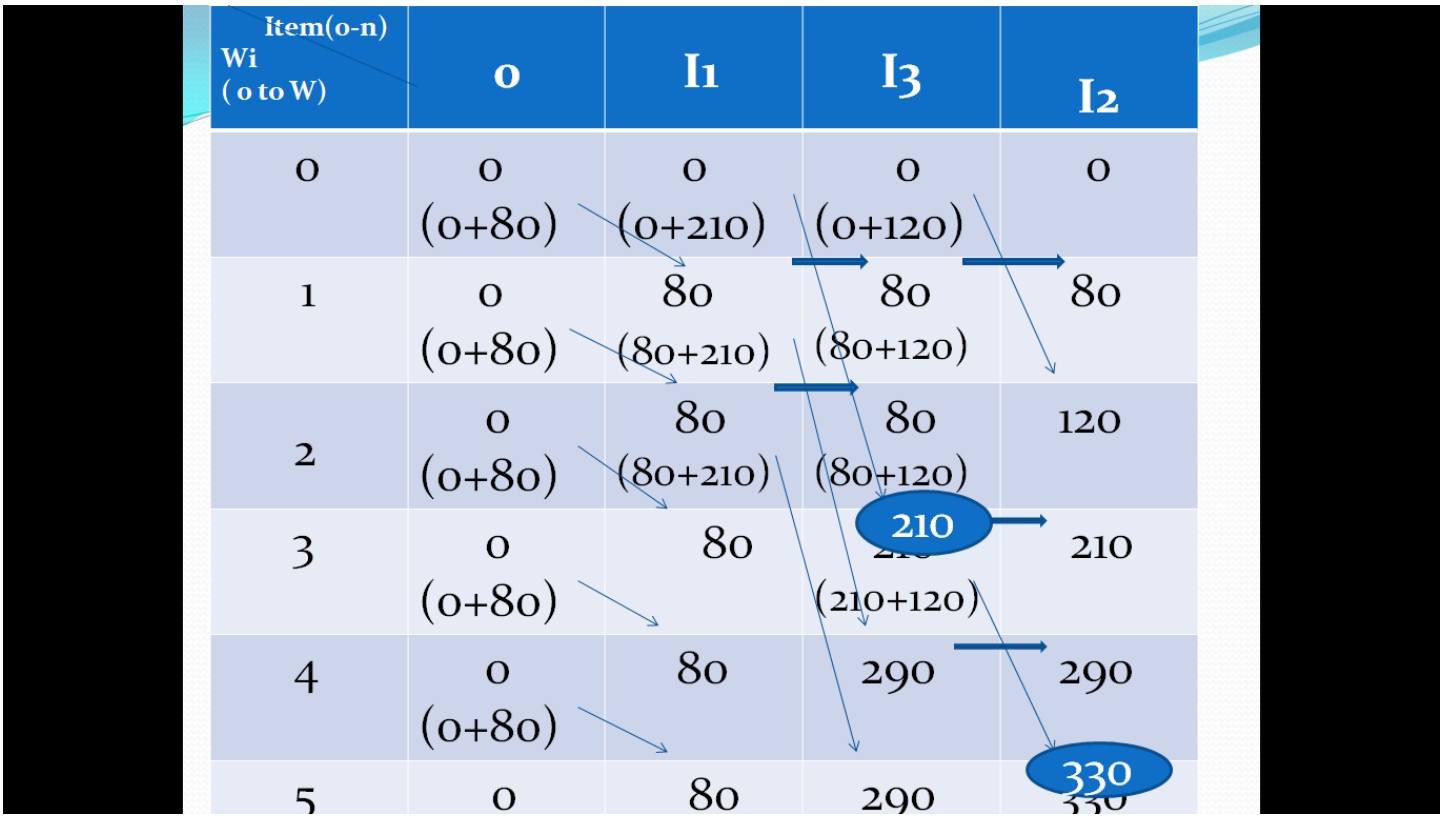
Ans.

```
if(wi<=w)
{
        if((bi+B[i-1][w-wi]) > (B[i-1][w]))
        {
                B[i][w]=bi+B[i-1][w-wi];
          }
else
{
B[i][w]=B[i-1][w];
}
```

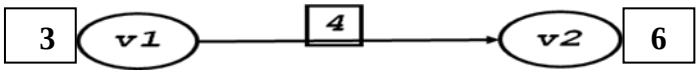
Decreasing order of price/weight item is taken in order by dividing weight price and capacity by 10 for easy tabular calculation.

1st row and 1st column initially filled by zero.

- Always choose the item which pointed out by the diagonal.
- In this example (I2+I3) is included.
- Total benefit=3300



Q.2 Define minimum spanning tree (MST). Find out key value of vertex v2 for prims and dijkstras algo. if current key values of vertices v1 & v2 and edge weight are given at a particular time during implementation of their procedure as shown in following figure:- [2+2 marks]



Ans. 2

Given an undirected, connected, weighted graph G=(V,E).
We wish to find an acyclic subset $T \subseteq E$ that connects all the vertices and whose total weight:

$$w(T)= \sum_{(u,v)\in T} w(u,v) \text{ is minimized.}$$

Where w(u,v) is the weight of edge (u,v).
T is called a minimum spanning tree of G.

For dijkstras key(V2)=6 since key(v1)+w(v1,v2)>key(v2)

and For prims key(V2)=4 since w(v1,v2)<key(v2)

Q.3 Write down algorithm for quick sort and Apply partition procedure of quick sort on the following elements to get the final position of pivot element (assume first element as pivot): [4+4=8 marks]
6,5,3,13,8,10,2,11

Ans 3.

```
void quicksort(int a[],int p,int q)
{
 int r;
if(p>=q)
 {
   printf("p>=q no op");
 }
 else   //(p<q)
 {
   r=partition(a,p,q);
     quicksort(a,p,r-1);
     quicksort(a,r+1,q);
  }
}
int partition(int a[],int p,int q)
{
   int x,i,j,temp;
   x=a[p];
   i=p;
   for(j=p+1;j<=q;j++)
   {
    if(a[j]<=x)
    {
     i=i+1;
    temp=a[i];
    a[i]=a[j];
    a[j]=temp;
    }

    a[p]=a[i];
    a[i]=x;
    return i;
}
```

p=0 & q=7,pivot=6 i=0 & j=1 a[j]<=pivot i=i+1=1 exch a[i]=5 & a[j]=5
6 5 3 13 8 10 2 11 i=1 & j=2 a[j]<=pivot i=i+1=2 exch a[i]=3 & a[j]=3

6 5 3 13 8 10 2 11 i=2 & j=3 a[j]>pivot j=j+1=4
6 5 3 13 8 10 2 11 i=2 & j=4 a[j]>pivot j=j+1=5
6 5 3 13 8 10 2 11 i=2 & j=5 a[j]>pivot j=j+1=6
6 5 3 13 8 10 2 11 i=2 & j=6 a[j]<=pivot i=i+1=3 exch a[i]=13 & a[j]=2
6 5 3 2 8 10 13 11 i=3 & j=7 a[j]>pivot j=j+1=8
6 5 3 2 8 10 13 11 exch pivot a[0]=6 & a[i]=a[3]=2
2 5 3 6 8 10 13 11 r=3 is partition index at value 6