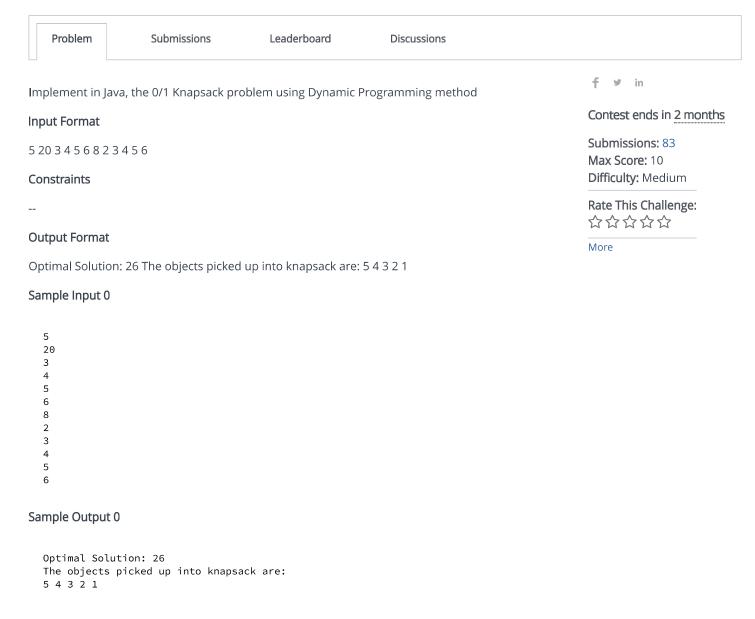
Q Search





All Contests > DAA\_LAB > Knapsack-Dynamic\_programming

## Knapsack-Dynamic\_programming



```
Java 7
1 //224G1A0553
2 ▼import java.util.Scanner;
3 √class DKnapsack {
    int c,n,p[],w[],v[][];
    public DKnapsack(int n,int c,int[] p,int[] w)
5
6 ▼ {
7
    super();
8
    this.n=n;
9
    this.c=c;
10
    this.w=w;
```

```
11 this.p=p;
12 ▼ this.v=new int[n+1][c+1];
13
   }
14 | void compute()
15 ₹ {
16 ▼ for(int i=0;i<=n;++i){
17 ▼ for(int j=0;j<=c;++j){
18 ▼ if(i==0||j==0){
19 ▼ v[i][j]=0;
20 }
21 ▼ else if(j-w[i]>=0)
22 ▼ {
23 ▼ v[i][j]=max(v[i-1][j],p[i]+v[i-1][j-w[i]]);
25 ▼ else if(j-w[i]<0)
26 ▼ {
27 ▼ v[i][j]=v[i-1][j];
28 }}}
29 ▼ System.out.println("Optimal Solution: "+v[n][c]);
30 traceback();
31
32 ▼ void traceback(){
   System.out.println("The objects picked up into knapsack are:");
33
34
    int i=n;
35
   int j=c;
36 while(i>0)
37 ▼ {
38 ▼ if(v[i][j]!=v[i-1][j])
39 ▼ {
    System.out.print(i+" ");
40
41 ▼ j=j-w[i];
42
    i--;
43
    }
44 ▼ else {
   i--;
45
46
   }}}
47 ▼ private int max(int i,int j){
    if(i>j)return i;
48
49
   else return j;
50
   }}
51 ▼public class KpDynamic{
52 ▼ public static void main(String[] args){
53
    int c,n;
54
   Scanner input=new Scanner(System.in);
    //System.out.println("Enter number of objects");
55
   n=input.nextInt();
56
57 ▼ int[] p=new int[n+1];
58 ▼ int[] w=new int[n+1];
59
60
    //System.out.println("Enter capacity of Knapsack");
61
    c=input.nextInt();
    //System.out.println("Enter profit for each "+n+" objects");
62
63
    for(i=1;i<=n;i++)
64 ▼ p[i]=input.nextInt();
    //System.out.println("Enter weight for each "+n+" objects");
65
    for(i=1;i<=n;i++)
66
67 ▼ w[i]=input.nextInt();
68
    DKnapsack dk=new DKnapsack(n,c,p,w);
69
    dk.compute();
70
    }}
```

Line: 70 Col: 3

<u>♣ Upload Code as File</u> Test against custom input

Run Code

Submit Code

```
Testcase 0 ✓
Congratulations, you passed the sample test case.
Click the Submit Code button to run your code against all the test cases.
Input (stdin)
 5
 20
 3
 4
 5
 6
 8
 2
 3
 4
 5
 6
Your Output (stdout)
 Optimal Solution: 26
 The objects picked up into knapsack are:
 5 4 3 2 1
Expected Output
 Optimal Solution: 26
 The objects picked up into knapsack are:
 5 4 3 2 1
```

Interview Prep | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy |