

EC4070: Data Structures and
Algorithms

Lab 08 -Greedy Algorithm

UPATHILAK A.L.A.A.

2020/E/165

GROUP C

SEMESTER 4

18th APR 2022

Problem

You are given a container full of water. Container can have limited amount of water. You also have N bottles to fill. You need to find the maximum numbers of bottles you can fill. Solve this problem using greedy approach.

Input: First line contains two integer, N and X, number of bottles and capacity of the container. Second line contains N space separated integers, capacities of bottles.

Output: Print the maximum number of bottles you can fill.

Sample Input:

5 10

8 5 4 3 2

Sample Output:

3

Implantation

```
import java.util.*;
```

```
public class Fill_bottel_2020e165
```

```
{
```

```
    public static int maximun_bottel(int []array,int capacity_of_container)
```

```
    {
```

```
        int total=0,number_of_bottel=0;
```

```
        int[] sorted_array=bubble_sort(array);
```

```
        if (capacity_of_container <= 0)
```

```
        {
```

```
            return 0;
```

```
        }
```

```
        for (int i = 0; i<sorted_array.length;++i)
```

```
        {
```

```
            if (total<=capacity_of_container)
```

```
            {
```

```
                total+=sorted_array[i];
```

```
                ++number_of_bottel;
```

```
            }
```

```
        }
```

```
        if(total>capacity_of_container)
```

```
        {
```

```

        number_of_bottel-=1;
    }
    return number_of_bottel;
}

public static int [] bubble_sort(int array[])
{
    int temp;
    for(int i=0;i<array.length-1;++i)
    {
        for(int j=0;j<array.length-i-1;++j)
        {
            if(array[j]>=array[j+1])
            {
                temp=array[j];
                array[j]=array[j+1];
                array[j+1]=temp;
            }
        }
    }
    return array;
}

public static void main(String arg[])
{
    System.out.print("Enter the number of bottel : ");
    Scanner Number_of_bottel=new Scanner(System.in);
    int number_of_bottel=Number_of_bottel.nextInt();
    System.out.print("Enter the  of capacity of the container : ");
    Scanner Capacity_of_container=new Scanner(System.in);
    int capacity_of_container=Capacity_of_container.nextInt();
    int []array_bottel=new int[number_of_bottel];
    System.out.print("Enter the bottel capacity: ");
    Scanner Bottels_capacities=new Scanner(System.in);
    for(int i=0;i<number_of_bottel;++i)
    {
        array_bottel[i]= Bottels_capacities.nextInt();
    }
}

```

```

    }
    System.out.println("Maixinum number of bottel : 
"+maximun_bottel(array_bottel,capacity_of_container));
}
}

```

```

1  import java.util.*;
2
3  public class Fill_bottel_2020e165
4  {
5      public static int maximun_bottel(int []array,int capacity_of_container)
6      {
7          int total=0,number_of_bottel=0;
8          int[] sorted_array=bubble_sort(array);
9          if (capacity_of_container <= 0)
10         {
11             return 0;
12         }
13         for (int i = 0; i<sorted_array.length;++i)
14         {
15             if (total<=capacity_of_container)
16             {
17                 total+=sorted_array[i];
18                 ++number_of_bottel;
19             }
20         }
21         if(total>capacity_of_container)
22         {
23             number_of_bottel-=1;
24         }
25         return number_of_bottel;
26     }
27
28     public static int [] bubble_sort(int array[])
29     {
30         int temp;
31         for(int i=0;i<array.length-1;++i)
32         {
33             for(int j=0;j<array.length-i-1;++j)
34             {
35                 if(array[j]>=array[j+1])
36                 {
37                     temp=array[j];
38                     array[j]=array[j+1];
39                     array[j+1]=temp;
40                 }
41             }
42         }
43         return array;
44     }
45
46     public static void main(String arg[])
47     {
48         System.out.print("Enter the number of bottel : ");
49         Scanner Number_of_bottel=new Scanner(System.in);
50         int number_of_bottel=Number_of_bottel.nextInt();
51         System.out.print("Enter the of capacity of the container : ");
52         Scanner Capacity_of_container=new Scanner(System.in);
53         int capacity_of_container=Capacity_of_container.nextInt();
54         int []array_bottel=new int[number_of_bottel];
55         System.out.print("Enter the bottel capacity: ");
56         Scanner Bottels_capacities=new Scanner(System.in);
57         for(int i=0;i<number_of_bottel;++i)
58         {
59             array_bottel[i]= Bottels_capacities.nextInt();
60         }
61         System.out.println("Maixinum number of bottel : "+maximun_bottel(array_bottel,capacity_of_container));
62     }
63 }

```

Figure 01:code

Output

```

C:\Users\Dell\OneDrive - University of Jaffna\Desktop\2020E_165_L8>javac Fill_bottel_2020e165.java
C:\Users\Dell\OneDrive - University of Jaffna\Desktop\2020E_165_L8>java Fill_bottel_2020e165
Enter the number of bottel : 5
Enter the of capacity of the container : 10
Enter the bottel capacity: 8 5 4 3 2
Maixinum number of bottel : 3

```

Figure 02:output

Conclusion

The problem of determining the most bottles that can be filled with a finite amount of water is solved by this code using a greedy algorithm method. The algorithm arranges the bottles according to their capacity and fills the container in decreasing order until it is full. By using the largest bottles first, this method makes sure that the most bottles can be filled. Overall, the code reads user input, sorts the bottles, and then calculates the most bottles that can be filled using the greedy method.

Discussion

Making the locally optimal solution at every step in the greedy algorithm is an easy and obvious problem-solving strategy with the goal of locating a global optimum. It is frequently applied to optimization issues where choosing the optimal choice from a range of options with immediate and quantifiable advantages is required. Although speedy and simple to use, greedy algorithms may not always produce the best results. Before using greedy plans, it is crucial to properly assess the issue and consider the implications of each option.