

Job Sequencing Problem

Problem is to schedule jobs so that profit is maximum. A class Job with data members job id, deadline and profit is given. We need to perform the jobs in such a way that the profit is maximised. So we sort jobs according to profit using a comparator

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

The approach is to complete the Job with highest profit first and mark that day as visited. Maintain a vector to keep track of which day is unoccupied. If we find a task whose deadline is less than the number of jobs scheduled, we skip it. Basically the idea is that if a task has a deadline of 6, we try to schedule it at the 6th slot. If it's not possible, we move in reverse from the deadline to find an empty slot.

When we find an empty slot for a job, we increase the number of jobs, add its profit to the answer, and break from the loop.

```
struct Job
{
    int id; // Job Id
    int dead; // Deadline of job
    int profit; // Profit if job is over before or on deadline
};
*/
bool comparison(Job a, Job b)
{
    return a.profit > b.profit;
}
class Solution
{
    //Function to find the maximum profit and the number of jobs done.
    public:
    vector<int> JobScheduling(Job arr[], int n)
    {
        // your code here
        sort(arr, arr+n, comparison);
        int maxi = arr[0].dead;
```

```

        vector<int> ans;
        for(int i=0;i<n;i++)
        {
            maxi=max(maxi,arr[i].dead);
        }
        vector<int>schedule(maxi+1,-1);
        int maxprofit=0,njobs=0;
        for(int i=0;i<n;i++)
        {
            for(int j=arr[i].dead;j>0;j--)
            {
                if(schedule[j]==-1)
                {
                    schedule[j]=arr[i].id;
                    njobs++;
                    maxprofit+=arr[i].profit;
                    break;
                }
            }
        }
        ans.push_back(njobs);
        ans.push_back(maxprofit);
        return ans;
    }
};

```

coding ninjas soln :

```

#include <bits/stdc++.h>

bool comparator(vector<int> a,vector<int> b)
{
    if(a[1]>b[1])
        return true;
    return false;
}
int jobScheduling(vector<vector<int>> &jobs)
{
    // Write your code here
    sort(jobs.begin(),jobs.end(),comparator);
    int maxi=0;
    for(int i=0;i<jobs.size();i++)
    {
        maxi=max(maxi,jobs[i][0]);
    }
    // cout<<maxi;
    int profit=0,schedule[maxi+1]={0};
    for(int i=0;i<jobs.size();i++)
    {

```

```
        for(int j=jobs[i][0];j>=1;j--)  
        {  
            if(schedule[j]==0)  
            {  
                profit+=jobs[i][1];  
                schedule[j]=i+1;  
                break;  
            }  
        }  
    }  
    return profit;  
}
```