GeeksforGeeks

A computer science portal for geeks

GeeksQuiz

- Home
- Algorithms
- DS
- GATE
- Interview Corner
- Q&A
- C
- C++
- Java
- Books
- Contribute
- Ask a Q
- About

Array
Bit Magic
C/C++
Articles
GFacts
Linked List
MCQ
Misc
Output
String
Tree
Graph

Job Sequencing Problem | Set 1 (Greedy Algorithm)

Given an array of jobs where every job has a deadline and associated profit if the job is finished before the deadline. It is also given that every job takes single unit of time, so the minimum possible deadline for any job is 1. How to maximize total profit if only one job can be scheduled at a time.

Examples:

```
Input: Four Jobs with following deadlines and profits
  JobID    Deadline      Profit
    a        4            20
    b        1            10
    c        1            40
    d        1            30
Output: Following is maximum profit sequence of jobs
```

```
        c, a


Input:  Five Jobs with following deadlines and profits
   JobID     Deadline     Profit
     a         2           100
     b         1           19
     c         2           27
     d         1           25
     e         3           15
Output: Following is maximum profit sequence of jobs
        c, a, e
```

We strongly recommend to minimize your browser and try this yourself first.

A **Simple Solution** is to generate all subsets of given set of jobs and check individual subset for feasibility of jobs in that subset. Keep track of maximum profit among all feasible subsets. The time complexity of this solution is exponential.

This is a standard Greedy Algorithm problem. Following is algorithm.

```
1) Sort all jobs in decreasing order of profit.
2) Initialize the result sequence as first job in sorted jobs.
3) Do following for remaining n-1 jobs
.......a) If the current job can fit in the current result sequence
          without missing the deadline, add current job to the result.
          Else ignore the current job.
```

The Following is C++ implementation of above algorithm.

```
// Program to find the maximum profit job sequence from a given array
// of jobs with deadlines and profits
#include<iostream>
#include<algorithm>
using namespace std;

// A structure to represent a job
struct Job
{
   char id;      // Job Id
   int dead;    // Deadline of job
   int profit;  // Profit if job is over before or on deadline
};

// This function is used for sorting all jobs according to profit
bool comparison(Job a, Job b)
{
     return (a.profit > b.profit);
}

// Returns minimum number of platforms reqquired
void printJobScheduling(Job arr[], int n)
{
    // Sort all jobs according to decreasing order of prfit
    sort(arr, arr+n, comparison);
    int result[n]; // To store result (Sequence of jobs)
    bool slot[n];  // To keep track of free time slots

    // Initialize all slots to be free
    for (int i=0; i<n; i++)
        slot[i] = false;

    // Iterate through all given jobs
    for (int i=0; i<n; i++)
    {
       // Find a free slot for this job (Note that we start
       // from the last possible slot)
       for (int j=min(n, arr[i].dead)-1; j>=0; j--)
       {
          // Free slot found
          if (slot[j]==false)
          {
             result[j] = i;  // Add this job to result
             slot[j] = true; // Make this slot occupied
             break;
          }
       }
    }

    // Print the result
    for (int i=0; i<n; i++)
       if (slot[i])
         cout << arr[result[i]].id << " ";
}

// Driver program to test methods
int main()
{
    Job arr[5] = { {'a', 2, 100}, {'b', 1, 19}, {'c', 2, 27},
                   {'d', 1, 25}, {'e', 3, 15}};
    int n = sizeof(arr)/sizeof(arr[0]);
    cout << "Following is maximum profit sequence of jobs\n";
    printJobScheduling(arr, n);
    return 0;
}
```

Output:

```
Following is maximum profit sequence of jobs
c a e
```

Time Complexity of the above solution is O(n^2). It can be optimized to almost O(n) by using union-find data structure. We will son be discussing the optimized solution.

Sources:
http://ocw.mit.edu/courses/civil-and-environmental-engineering/1-204-computer-algorithms-in-systems-engineering-spring-2010/lecture-notes/MIT1_204S10_lec10.pdf

This article is contributed by **Shubham**. Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

Related Topics:

- Linearity of Expectation
- Iterative Tower of Hanoi
- Count possible ways to construct buildings
- Build Lowest Number by Removing n digits from a given number
- Set Cover Problem | Set 1 (Greedy Approximate Algorithm)
- Find number of days between two given dates
- How to print maximum number of A's using given four keys
- Write an iterative O(Log y) function for pow(x, y)

Tags: Greedy Algorithm

Tweet 3 g+1 1

Writing code in comment? Please use **ideone.com** and share the link here.

20 Comments **GeeksforGeeks** 1 **Login**

Recommend Share Sort by Newest

Join the discussion…

John Smith the Mormon • 2 months ago

It is better to first sort them by increasing deadlines and as a secondary sort (for those with equal deadlines) based on decreasing profit. The end result is the same, but we can formalize the problem better this way.

• Reply • Share ›

Tareq Obaida • 2 months ago

Pseudo code is not clear enough. Let me try if I can make it clear..

```
1. Sort all jobs in decreasing order of profit. let jobs is the sorted array.
2. let result[1..n] and timeslot[1...highestdeadline] be two arrays
3.initialize all element of timeslot="unoccupied"
4.for all job in jobs
5. for j=job.deadline to 1
6. if timeslot[j]="unoccupied"
7 result=result union {job}
8 timeslot[j]="occupied"
9. break
10. return result
```

1 ^ | ˅ • Reply • Share ›

Aditi Rai • 2 months ago

after sorting the job in decreasing order of their profit... can't we do like this???

```
int i,j,t;
for (i = 1; i <= n; i++) {
if (time[i] != 0) {
t = time[i];
printf("%c ",job[i]);

for (j = i+1; j <= n; j++) {
if (t > time[j])
time[j] = 0;

else if (t <= time[j])
time[j]--;
}
time[i]=0;
}
}
```

please reply....!!!!!

^ | ˅ • Reply • Share ›

Guest • 3 months ago

Use pair, first sort in increasing order of deadline then for particular deadline sort profit in decreasing order. Select first job of each deadline.

^ | ˅ • Reply • Share ›

bava • 3 months ago

give the java code for job sequence with dead line fro analysis purpose

^ | ˅ • Reply • Share ›

Ankit Chaudhary • 3 months ago

Do it in simple way :

Take a variable time=1, and inside first for loop

do this : `if(arr[i].dead>=time){ result[k++]=i; time++;}`

where k = current length of result array.

to print answer :

```
for(int i=0;i<k;i++) printf("%c="" ",arr[result[i]].id);="">
```

^ | ˅ • Reply • Share ›

Reply • Share ›

Guest • 3 months ago

Can't we do like this in nlogn

```
void printJobScheduling(Job arr[], int n)

{

// Sort all jobs according to decreasing order of prfit

sort(arr, arr+n, comparison);

int result[n]; // To store result (Sequence of jobs)

// Iterate through all given jobs

int k=0;

for (int i=0,time=1; i<n; i++)="" {="" if(arr[i].dead="">= time){

result[k++]=i;

t++;

}

}

// Print the result

for (int i=0; i<k; i++)="" cout="" <<="" arr[result[i]].id="" <<="" "="" ";="" }="">
```

Reply • Share ›

Siddharth Singh • 3 months ago

In That Example One , Shouldn't The Answer be b,c,d Producing The Maximum Profit of 70 With Deadline 3 Than 4 ??

Reply • Share ›

spock • 3 months ago

```
import java.lang.Comparable;

public class JobScheduling{
public static class Job implements Comparable{
int deadline;
int profit;
public Job(int deadline, int profit){
this.deadline = deadline;
this.profit = profit;
this.profit = profit;
}
@Override
public int compareTo(Object job2){
if(job2 == null){
return -1;
}
if(!(job2 instanceof Job)){
return -1;
}
```

see more

˄ | ˅ • Reply • Share ›

spock • 4 months ago

Couldn't understand the question, How does the jobs c & a are chosen.

3 ˄ | ˅ • Reply • Share ›

Sumit Kesarwani • 4 months ago

http://www.cs.sfu.ca/~kabanets...

1 ˄ | ˅ • Reply • Share ›

Prakash • 4 months ago

second case has solution d,a,e. Correction required.

1 ˄ | ˅ • Reply • Share ›

Ajay Gaur • 5 months ago

This can be done in O(nlogn) by sorting by deadline and arranging the benefit to their corresponding deadline (modifying the quick-sort), then, by iterating over the deadline, choosing the most benefit for a deadline.

˄ | ˅ • Reply • Share ›

Ajay Gaur • 5 months ago

In the second example, should not the sequence be <d,a,e>

˄ | ˅ • Reply • Share ›

abhi • 5 months ago

will it always give optimal solution,
i dont thing it will work in all cases.

˄ | ˅ • Reply • Share ›

poorna • 6 months ago

isnt the time complexity of generating a subsets is 2^n?

˄ | ˅ • Reply • Share ›

Anurag Singh ➦ poorna • 6 months ago

I guess you are talking about the complexity of implementation provided above.
The implementation provided above is not based on simple solution, which checks all subsets.
It's based on Greedy Approach which takes O(n^2).
Simple solution explained above which checks all subsets will take O(2^n).

⌃ | ⌄ • Reply • Share ›

St • 6 months ago

We can do it in O(nlogn). Sort array by deadline. Create a max heap (comparison by profit). Start from the max deadline and put everything with the same deadline to the heap. Element in the top of the heap add to the result. Then add to the heap all elements with the next deadline (going in decreasing order). Then add to the result from the top of the heap. And so on till the deadline 1.

8 ⌃ | ⌄ • Reply • Share ›

Gravity • 6 months ago

Can you explain the optimization using union-find data structure ?

⌃ | ⌄ • Reply • Share ›

GeeksforGeeks Mod ➦ Gravity • 6 months ago

We will soon be covering it in set 2.

1 ⌃ | ⌄ • Reply • Share ›

✉ Subscribe   Ⓓ Add Disqus to your site   Privacy

-
-
-
-
    - Interview Experiences
    - Advanced Data Structures
    - Dynamic Programming
    - Greedy Algorithms
    - Backtracking

- Pattern Searching
- Divide & Conquer
- Mathematical Algorithms
- Recursion
- Geometric Algorithms

Popular Posts

- All permutations of a given string
- Memory Layout of C Programs
- Understanding “extern” keyword in C
- Median of two sorted arrays
- Tree traversal without recursion and without stack!
- Structure Member Alignment, Padding and Data Packing
- Intersection point of two Linked Lists
- Lowest Common Ancestor in a BST.
- Check if a binary tree is BST or not
- Sorted Linked List to Balanced BST

Follow @GeeksforGeeks

Recent Comments

- lt_k

  i need help for coding this function in java...

  Java Programming Language · 1 hour ago

- Piyush

  What is the purpose of else if (recStack[*i])...

  Detect Cycle in a Directed Graph · 1 hour ago

- Andy Toh

  My compile-time solution, which agrees with the...

  Dynamic Programming | Set 16 (Floyd Warshall Algorithm) · 1 hour ago

- lucy

  because we first fill zero in first col and...

  Dynamic Programming | Set 29 (Longest Common Substring) · 2 hours ago

- lucy

  @GeeksforGeeks i don't n know what is this long...

Dynamic Programming | Set 28 (Minimum insertions to form a palindrome) · 2 hours ago

- manish

  Because TAN is not a subsequence of RANT. ANT...

  Given two strings, find if first string is a subsequence of second · 2 hours ago

-