

UCS415 Design and Analysis of Algorithms

Lab Assignment 2: Greedy Approach

1. Given the Input: $\text{start[]} = \{1, 3, 0, 5, 8, 5\}$, $\text{finish[]} = \{2, 4, 6, 7, 9, 9\}$; using Activity selection. Select the maximum number of activities that can be performed by a single person, assuming that a person can only work on a single activity at a time.
2. We are given two arrays that represent the arrival and departure times of trains, the task is to find the minimum number of platforms required so that no train waits.

Input: $\text{arr[]} = \{9:00, 9:40, 9:50, 11:00, 15:00, 18:00\}$, $\text{dep[]} = \{9:10, 12:00, 11:20, 11:30, 19:00, 20:00\}$

Output: 3

3. Given an array of jobs where every job has a deadline and associated profit if the job is finished before the deadline, maximize the total profit if only one job can be scheduled at a time.

JobId	Deadline	Profit
A	4	20
B	1	10
C	1	40
D	1	30

4. Given the weights and profits of N items, in the form of {profit, weight}, Input: $\text{arr[]} = \{\{60, 10\}, \{100, 20\}, \{120, 30\}\}$, put these items in a knapsack of capacity $W = 50$ to get the maximum total profit in the knapsack. Use Fractional Knapsack, and break items for maximizing the total value of the knapsack.
5. Huffman Coding is a lossless data compression algorithm. The algorithm assigns variable-length codes to input characters, with shorter codes assigned to more frequent characters. Write a program to implement Huffman Coding. Given a string as input, your task is to:
 - a. Build a Huffman Tree for the characters in the string.
 - b. Generate the corresponding Huffman Codes for each character.
 - c. Encode the string using the generated Huffman Codes.
 - d. Decode the encoded string back to its original form.

Additional Questions:

1. <https://www.interviewbit.com/problems/majority-element/>
2. <https://www.interviewbit.com/problems/distribute-candy/>
3. <https://www.hackerearth.com/practice/algorithms/greedy/basics-of-greedy-algorithms/tutorial/> [Solve the problem available at the end]
4. <https://www.codechef.com/problems/SUBSEG2>
5. <https://www.codechef.com/problems/FGFS/>
6. <https://www.hackerrank.com/challenges/board-cutting/problem>