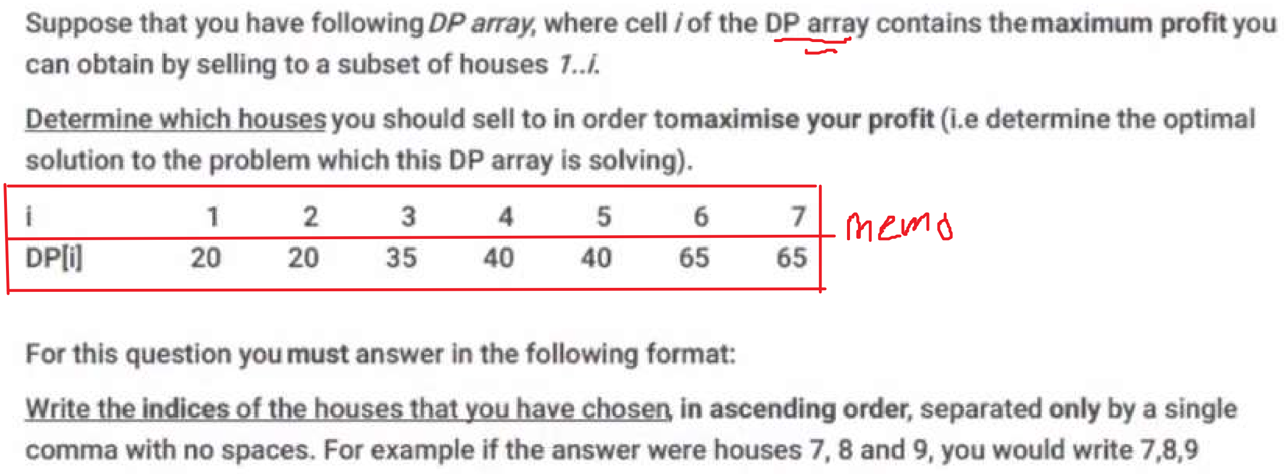
Past Year DP

Tuesday, 31 May, 2022 01:58



Suppose that you have following *DP array*, where cell *i* of the DP array contains the **maximum profit** you can obtain by selling to a subset of houses *1..i*.

2 marks

Determine which houses you should sell to in order to **maximise your profit** (i.e determine the optimal solution to the problem which this DP array is solving).

i	1	2	3	4	5	6	7
DP[i]	20	20	35	40	40	65	65

memo

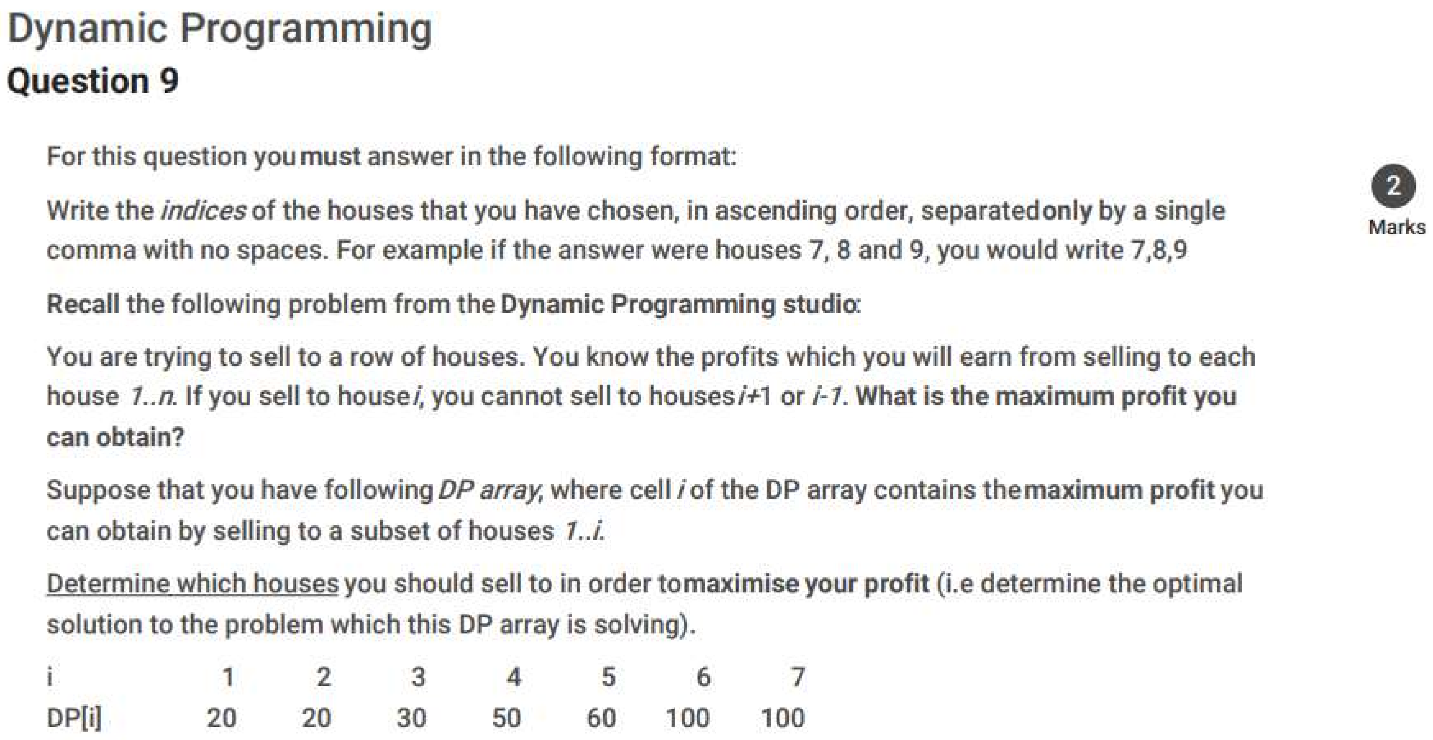
For this question you **must** answer in the following format:

Write the indices of the houses that you have chosen, **in ascending order**, separated **only** by a single comma with no spaces. For example if the answer were houses 7, 8 and 9, you would write 7,8,9

1,4,6

Sample Paper 2022 Sem 1

Wednesday, 1 June, 2022 00:49



Dynamic Programming

Question 9

2 Marks

For this question you **must** answer in the following format:

Write the *indices* of the houses that you have chosen, in ascending order, separated **only** by a single comma with no spaces. For example if the answer were houses 7, 8 and 9, you would write 7,8,9

Recall the following problem from the **Dynamic Programming studio**:

You are trying to sell to a row of houses. You know the profits which you will earn from selling to each house *1..n*. If you sell to house *i*, you cannot sell to houses *i*+1 or *i-1*. **What is the maximum profit you can obtain?**

Suppose that you have following *DP array*, where cell *i* of the DP array contains the **maximum profit** you can obtain by selling to a subset of houses *1..i*.

Determine which houses you should sell to in order to **maximise your profit** (i.e determine the optimal solution to the problem which this DP array is solving).

i	1	2	3	4	5	6	7
DP[i]	20	20	30	50	60	100	100

1,4,6

Studio07 Q2

Wednesday, 13 April, 2022 00:29

Problem 2. Suppose that you are a door-to-door salesman, selling the latest innovation in vacuum cleaners to less-than-enthusiastic customers. Today, you are planning on selling to some of the n houses along a particular street. You are a master salesman, so for each house, you have already worked out the amount c_i of profit that you will make from the person in house i if you talk to them. Unfortunately, you cannot sell to every house, since if a person's neighbour sees you selling to them, they will hide and not answer the door for you. Therefore, you must select a subset of houses to sell to such that none of them are next to each other, and such that you make the maximum amount of money.

For example, if there are 10 houses and the profits that you can make from each of them are 50, 10, 12, 65, 40, 95, 100, 12, 20, 30, then it is optimal to sell to the houses 1, 4, 6, 8, 10 for a total profit of \$252. Devise a dynamic programming algorithm to solve this problem, i.e. to return the maximum profit you can obtain.

(a) Describe in plain English, the optimal substructure present in the problem.

(b) Define a set of overlapping subproblems that are based on the optimal substructure.

(c) What are the base case subproblems and what are their values?

(d) Write a recurrence relation that describes the solutions to the subproblems.

(e) Write pseudocode that implements all of this as a dynamic programming algorithm.

a) We can start by checking the optimal solution for each i by checking i+1 and i-1. We don't need to care too much about the later i by comparing i+1, we can just focus on the optimal solution that is behind i-1 and i-2. Either we include the current i or exclude it and get the optimal solution of i-2.

b) The overlapping sets would be the previous i-2 or i-1 checking for example.

	0	1	2	3	4	5	6	7	8	9	10	11	12
houses	Ø	50	10	12	65	40	95	100	12	20	30	11	
memo	0	50	50	62	115	115	210	215	222	235	252	252	
solutions	Ø	Ø 1	Ø 1	Ø 1 3	Ø 1 4	Ø 1 4	Ø 1 4 6	Ø 1 4 7	Ø 1 4 6 8	Ø 1 4 7 9	Ø 1 4 6 8 10	Ø 1 4 6 8 10	

As u can see that there are overlapping subproblem.

c) Base case would be an empty set. Value would be 0

d) Recurrence relation :

base case

$$\text{memo}[i] = \begin{cases} 0 & ; i = 0 \\ \text{include} : \text{memo}\{i-2\} + \text{house}\{i\} & ; i \geq 0 \\ \text{exclude} : \text{memo}\{i-1\} & ; i \geq 0 \end{cases}$$

e) 1) Initializing base case to 0
2)For loop i from 1 to N:
3)Set optimal value of i from max(value[i-1],value[i-2] + value[i])
4)return value