

Solutions for Data Structures and Algorithms Spring 2023 — Problem Sets

By Dmitriy Okoneshnikov, B22-DSAI-04

February 21-22, 2023

Week 5. Problem set

The National Health Council wants to improve health care in three of the most underdeveloped regions. Currently it has five medical teams available to allocate among such regions to improve their medical care, health education, and training programs. Therefore, the council needs to determine how many teams (if any) to allocate to each of these regions to maximize the total effectiveness of the five teams. The teams must be kept intact, so the number allocated to each region must be an integer.

The measure of performance being used is additional person-years of life. (For a particular region, this measure equals the increased life expectancy in years times the region's population.) The following table gives the estimated additional person-years of life (in multiples of 1,000) for each region for each possible allocation of medical teams.

1. Describe a general algorithm for any number M of available medical teams, any number R of regions and table E with estimates:

- (a) Summarize the idea for a naive recursive algorithm.

Answer.

The idea for a naive recursive algorithm is simple: we need to brute-force every allocation of teams through the regions. It is obvious that we need to allocate all the teams to get maximal effectiveness. So we have to find all allocations $\sum_{i=1}^R M_i = M$, where M_i is number of teams in region i , and find the one with the maximal effectiveness by accessing the table E.

The base case for this algorithm will be when we have allocated all the teams into the regions. In the base case we will return the total additional person-years of life if we use this allocation.

Then we add one more team to a region and initiate the recursion call with this new allocation. We will have to repeat this process for all other regions.

- (b) Identify overlapping subproblems.

Answer.

The overlapping subproblems appear because we can get to the same allocation of teams in different ways and, therefore, counting the total effectiveness of this allocation several times.

- (c) Write down pseudocode for the dynamic programming algorithm that solves the problem (top-down or bottom-up). It is enough to compute maximum performance without specifying team allocation.

Answer.

Let's add memoization to our naive recursive algorithm (top-down approach):

```

1 Memoized-Team-Allocation( $M, R, E$ )
2   let  $dp[0:R, 0:M]$  be a new table //  $dp[i, j]$  is the maximum additional
3   // person-years of life that we can get by allocating  $j$  medical teams
4   // in the first  $i$  regions
5   for  $i = 1$  to  $R$ :
6     for  $j = 1$  to  $M$ :
7       for  $k = 0$  to  $j$ :
8          $dp[i, j] = \max\{dp[i, j], dp[i-1, j-k] + E[k, i-1]\}$ 
9   return  $dp[R, M]$ 

```

2. Provide asymptotic worst-case time complexity with justification

(a) for the naive recursive algorithm

Solution.

Due to our base case and the fact that we add only one team a time, the R -ary (as we call our function R times from one recursion call) recursion tree has a height of M . Therefore, the maximum number of nodes is R^M . The base case is $O(R)$, so the total worst-time complexity is $O(R^M \cdot R) = O(R^{M+1}) = O(R^M)$ 🙄.

Answer. $O(R^M)$

(b) for the dynamic programming algorithm

Solution.

It is easy to see that we have three nested loops going from 1 to R , 1 to M and 1 to M (worst case), therefore, the worst-time complexity is $O(RM^2)$.

Answer. $O(RM^2)$

3. Apply the dynamic programming algorithm to an instance of the problem below. You must provide the table with solutions for subproblems that are computed in the algorithm, as well as give the final answer to the problem.

Number of Teams	Region A	Region B	Region C
0	0	0	0
1	45	50	20
2	70	70	45
3	90	80	75
4	105	100	110
5	120	130	150

Solution.

i	j	dp						
1	1	0	0	0	0	0	0	0
		0	45	0	0	0	0	0
		0	0	0	0	0	0	0
		0	0	0	0	0	0	0
1	2	0	0	0	0	0	0	0
		0	45	70	0	0	0	0
		0	0	0	0	0	0	0
		0	0	0	0	0	0	0

i	j	dp					
1	3	0	0	0	0	0	0
		0	45	70	90	0	0
		0	0	0	0	0	0
		0	0	0	0	0	0
1	4	0	0	0	0	0	0
		0	45	70	90	105	0
		0	0	0	0	0	0
		0	0	0	0	0	0
1	5	0	0	0	0	0	0
		0	45	70	90	105	120
		0	0	0	0	0	0
		0	0	0	0	0	0
2	1	0	0	0	0	0	0
		0	45	70	90	105	120
		0	50	0	0	0	0
		0	0	0	0	0	0
2	2	0	0	0	0	0	0
		0	45	70	90	105	120
		0	50	95	0	0	0
		0	0	0	0	0	0
2	3	0	0	0	0	0	0
		0	45	70	90	105	120
		0	50	95	120	0	0
		0	0	0	0	0	0
2	4	0	0	0	0	0	0
		0	45	70	90	105	120
		0	50	95	120	140	0
		0	0	0	0	0	0
2	5	0	0	0	0	0	0
		0	45	70	90	105	120
		0	50	95	120	140	160
		0	0	0	0	0	0
3	1	0	0	0	0	0	0
		0	45	70	90	105	120
		0	50	95	120	140	160
		0	50	0	0	0	0
3	2	0	0	0	0	0	0
		0	45	70	90	105	120
		0	50	95	120	140	160
		0	50	95	0	0	0
3	3	0	0	0	0	0	0
		0	45	70	90	105	120
		0	50	95	120	140	160
		0	50	95	120	0	0
3	4	0	0	0	0	0	0
		0	45	70	90	105	120
		0	50	95	120	140	160
		0	50	95	120	140	0

i	j	dp					
3	5	0	0	0	0	0	0
		0	45	70	90	105	120
		0	50	95	120	140	160
		0	50	95	120	140	170

Answer. 170

References

- [1] T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein. *Introduction to Algorithms, Fourth Edition*. The MIT Press 2022