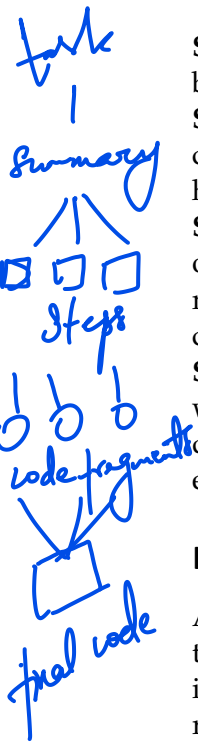
Week 3: Algorithmic Thinking I

Session 5: The Four Steps of Algorithmic Thinking



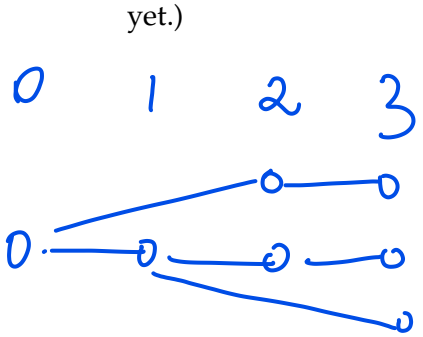
Step 1. Understand: Summarize the task in your own words and verify your understanding by manually computing the results for a few inputs.
Step 2. Decompose: Write clear and precise instructions for another human being to manually compute the appropriate results for any possible input, imagining that the person does not have access to the problem description but only has your instructions to go on.
Step 3. Analyze: For each part of the instructions above, plan how you would carry them out using Python. For the trickest parts, write fragments of runnable Python code to implement them, while creating sample intermediate inputs to test each fragment separately on the computer.
Step 4. Synthesize: Following the instructions from Step 2 and the code fragments from Step 3, write complete Python code to implement the instructions and solve the problem. You should do this in an incremental fashion and print intermediate outputs as you go to make sure that each part of the code matches your expectations.

In-class Exercise: Referral Marketing

A start-up's marketing team is trying to predict the growth of a new product launch. Assuming that each person who buys the product will refer a friend (who hasn't seen the product before) in the next month, as well as two months after buying, but will stop referring after that. Each referred friend will buy the product upon hearing about it and follow the same pattern of referral. Assume that

- each customer only buys the product once;
- there is one person who buys the product in month zero;
- the only buyers of the product are those referred, as described above.

How many people would buy the product in month 12? (You don't need to write any code yet.)



Therefore, buyers in one month are sum of buyers in previous two months.

Months	0	1	2	3	4	5	6	7	8	9	10	11	12
Buyers	1	1	2	3	5	8	13	21	34	55	89	144	233

Exercise 3.1 Python Code for Referral Marketing

Download the Jupyter notebook attached to this exercise on Blackboard and submit it there after completing it.

The notebook asks you to write Python code to compute the number of buyers in month 12, using the four steps of algorithmic thinking when developing the code. Since the focus of this week is the thinking process, it is important that you follow the steps, rather than try to code everything at once.

Sample Problem: Epidemic Capacity Planning

A city is trying to predict how many ventilators it will need during a certain epidemic. To help the policy makers analyze various scenarios, write a function called **capacityNeeded** with one input parameter:

- **arrivals**: a list of forecasted number of arrivals each week of new patients requiring a ventilator.

Assume that each patient requires a ventilator for exactly 3 weeks. The function should return an integer corresponding to the minimum number of ventilators needed to satisfy all the demand.

For example, if `arrivals=[5,8,3,10,7,4,9,5,8]`, then the function should return 22, because in the last three weeks, 9+5+8=22 ventilators are needed, and having 22 is sufficient to satisfy demand in any week. (In the first week, only 5 ventilators are needed; in the second week, 5+8=13 are needed; in the third week 5+8+3=16 are needed; in the fourth week 8+3+10=21 are needed; in the fifth week, 3+10+7=20 are needed, etc.)

In-class exercise: Practicing Steps 1 and 2

Manually compute the desired answer corresponding to the sample input `arrivals=[5,8,3,10,7,4,9,5,8]`. Ideally, you should create a table to track the computations, and describe precisely how each entry in the table is to be computed. This logic will be the basis of your coding later, but you don't need to write any code yet.

Exercise 3.2 Python Code for Epidemic Capacity Planning

Download the Jupyter notebook attached to this exercise on Blackboard and submit it there after completing it.

The notebook asks you to apply Steps 3 and 4 of algorithmic thinking to implement the instructions as described in the sample solutions to the in-class exercise, to solve the sample problem "Epidemic Capacity Planning."