

## Week 3: Algorithmic Thinking I

### Session 6: Translating English Steps into Working Code

#### In-Class Exercise: Paper Coding

One tip for efficiently solving a complex programming task is to first draft the code on paper. Perhaps counterintuitively, it is often faster if you first code on paper and then type it in, rather than if you worked directly on the computer to begin with. This is because hand writing helps you to focus on the overall logic rather than the syntax.

**Without using a computer, write on a piece of paper Python code that implements the following logic:** Given a list named `curVal`, representing the valuation (willingness to pay) of a customer for two products, as well as list named `priceVector`, representing the price of the two products, print "Purchase Product 0" if the customer purchases the first product; print "Purchase Product 1" if the customer purchases the second product; print "Purchase nothing" if the customer purchases neither.

**Assumptions:** The customer will purchase at most one product, and the product that is purchased is one in which the customer's willingness to pay is at least equal to the price. If the customer's valuations for both products are greater than or equal to the corresponding prices, then the customer will purchase the product in which the difference between his/her valuation and the price is the largest. If there is a tie in this difference, then the customer will purchase Product 0. If the valuation is less than the price for both products, then the customer would not purchase anything.

For example, if the valuation of a customer is `[9, 8]` then

- If `priceVector=[6, 4]`, then the customer will purchase Product 1 because  $8 - 4 > 9 - 6$ .
- If `priceVector=[5, 4]`, then the customer will purchase Product 0 because  $9 - 5 \geq 8 - 4$ .
- If `priceVector=[10, 8]`, then the customer will purchase Product 1.
- If `priceVector=[10, 10]`, then the customer will purchase nothing.

For concreteness, you can start your code with the following two lines.

```
curVal=[25,15]
priceVector=[25,10]
```

```
if curVal[0] > priceVector[0] or curVal[1] > priceVector[1]:
    if priceVector[0] - curVal[0] == priceVector[1] - curVal[1]:
        print 'Purchase product 0'
    else:
        purchase = max(priceVector[0] - curVal[0], priceVector[1] - curVal[1])
        idn = priceVector.index(purchase)
        print f'Purchase product {idn}'
else:
    print 'Purchase nothing'
```

1

## Sample Problem: Demand Estimation for Substitutable Products

Write a function named "demand" with two input arguments:

- priceVector: a list of length 2 containing two positive numbers, corresponding to the proposed prices for the two products.
- values: a list in which each element is a list of length 2, corresponding to the valuation of a customer for the two products.

The function should return a list of two numbers, representing the number of customers purchasing each product. Assume the same customer behavior as in the paper coding exercise.

**Sample run:**

```
values=[[25,15],[18,18],[30,20],[30,30]]
priceVector=[25,20]
demand(priceVector,values)
```

**Correct result:**

[2, 1]

### In-Class Exercise: Complete Steps 1 and 2 for the above Sample Problem

**Step 1. Understand:** Summarize the task in your own words and verify your understanding by manually computing the results for a few inputs.

*Given two products, find purchasers for each given that a customer buys a product if their evaluation is more than product cost.*

**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.

- *Read the evaluation for each customer one at a time.*
- *Compare evaluation of customer & product cost to compute difference between the two for each position.*
- *If differences > 0 then customer buys product w/ greatest difference*
- *Otherwise customer buys nothing.*
- *If differences are same then customer will buy first product.*

### Exercise 3.3: Python Code for Demand Estimation

Download the Jupyter notebook attached to this exercise on Blackboard and submit it there after completing it. The notebook asks you to complete steps 3 and 4 of the sample problem above.