DSA Homework 1

Aryan Rudraraju

November 18, 2025

1.1

a) Sorting Algorithms in Order Book Management

Sorting algorithms may be used for order book management. Market makers need to know all buy and sell orders for a particular asset. These need to be sorted as, whenever someone chooses to cross the bid ask spread, a sale must happen. However, the order must go to the person with the highest bid or the lowest selling price. The way to ensure this happens is by using sorting algorithms.

b) NPV (Net Present Value) Calculation Complexity

The formula for **Net Present Value (NPV)** is given by:

$$NPV = \frac{R_t}{(1+i)^t}$$

Where:

- R_t is the net cash flow at time t,
- i is the discount rate,
- t is the time of the cash flow.

The expected complexity for an NPV calculation should be **O(n)**, as it is calculating n cash flows which are linear.

Asymptotic Analysis

Asymptotic analysis (big-O) is more useful early on in the project, as it is quick to work out, and doesn't require the algorithm to be built out. It allows you to choose which algorithms to develop further easily without having to code them. To find big-O complexity, you can count the number of operations that are being performed.

Empirical Analysis

Once the algorithm is developed, empirical analysis is more useful. It can take into account other factors that may not be as easy with asymptomatic analysis, such as memory efficiency. It is also more accurate, as big-O may have some scalar differences that could be less obvious than empirical analysis will reveal.

1.2

a) Complexity of Computing Average Return for One Asset

Complexity of computing the return of one asset:

- Compute daily returns for d days, which has the complexity $O(d)$.
- Sum the d daily returns, which is also $O(d)$.
- Divide by d to obtain an average, which is $O(1)$.

Hence, the overall complexity of computing the average return for one asset is $O(d)$.

b) Finding the Asset with the kth Smallest Average Return

To find the asset with the kth smallest average return over all assets:

- First, calculate the average return for all n assets, which will have complexity $O(d)$ for one asset. This means the complexity for all n assets is $O(nd)$.
- Then, every return needs to be sorted. Sorting all n average returns takes $O(n \log n)$.

Thus, the overall complexity is:

$$O(nd) + O(n \log n) = O(nd + n \log n)$$

c) Complexity Lower Bound for Finding the kth Smallest Average Return

The problem will always be at least $O(n)$, as to find the kth smallest average return, it's necessary to look at all n assets to calculate their average returns. Without calculating all the average returns, it would be impossible to find the kth smallest asset. Hence, stating that the complexity is **at least** $O(n)$ is meaningless.

Furthermore, big-O is typically used as an upper bound for complexity. Stating a lower bound for complexity is not as useful as knowing the maximum complexity that could hypothetically occur.

1.3

a) Complexity of Allocating Items in Auction

As the quantity of items and bidders increases, finding a suitable allocation becomes exponentially harder. The number of possible allocations increases with the binomial coefficient $\binom{n}{m}$. Additionally, not all bidders may have linear valuation functions. They may have diminishing marginal utility, or other more complex utility functions.

b) Approach with Concave Valuation Functions

Firstly, calculate a marginal value for every bidder, which represents the value they would get from receiving one more item. As the valuation functions are concave, the marginal value decreases as x increases.

Then, sort the bidders based on their greatest marginal value to the least. Allocate an item to the bidder with the highest marginal value, then recalculate the marginal value for that bidder. Re-sort the bidders by marginal value and allocate the next item to the bidder with the highest remaining marginal value. Repeat this process until all items are allocated.