1.1-1

Describe your own real-world example that requires sorting. Describe one that requires finding the shortest distance between two points.

My real- world is a grocery store that needs to organize its products on shelves based on their expiration dates of the product. The products with the earliest expiration dates would come first on the shelf hence the older products are consumed before they expire, reducing waste and minimizing the risk of foodborne illnesses. I would personally use an insertion algorithm

Finding the shortest distance between two points, take for instance a grocery store delivery driver needs to find the most efficient route to deliver packages to multiple addresses. Considering factors like road distance and traffic I would use Dijkstra's algorithm.

1.1-2

Other than speed, what other measures of efficiency might you need to consider in

a real-world setting?

In real-world settings, I would consider the memory space in my case shelf space how much space does the algorithm need

I would also consider the input size of the algorithm which will later determine the time the algorithm takes to complete excitation.

1.1-3

Select a data structure that you have seen, and discuss its strengths and limitations.

The strengths

Linked lists can easily grow or shrink without requiring pre -allocation of memory that is to say its Dynamic in size.

Elements can be inserted or deleted at any position in the list without shifting the entire array.

Linked lists don't suffer from memory fragmentation, as elements can be stored anywhere in memory.

The limitations

Accessing (random access) an element at a specific index requires traversing the list from the beginning, which can be inefficient for large lists.

Each element in a linked list requires additional space for the pointer to the next element.

Iterating backwards through a linked list is less efficient than iterating forwards, as it requires traversing the list from the end to the beginning.

1.1-4

How are the shortest-path and traveling-salesperson problems given above similar?

How are they different?

Similarity

Both the shortest-path and traveling-salesperson involve finding optimal paths between points in a graph.

Difference

The shortest-path finds the shortest path between two specific points while a traveling salesperson finds the shortest path that visits all nodes exactly once.

The shortest-path problems are generally easier to solve while the traveling salesperson problem, which is NP-hard.

1.1-5

Suggest a real-world problem in which only the best solution will do. Then come

up with one in which <approximately= the best solution is good enough.

The real-world problem is analyzing real-time stock market data to make trading decisions.

Case one is Entire Input Available

A financial analyst wants to analyse a historical dataset of stock prices to identify trends and patterns.

Input would be the entire historical dataset of stock prices is available for analysis.

Case two is Incremental Input

A day trader is monitoring the stock market in real-time to make quick trading decisions.

Input would be new stock price data arrives continuously, requiring the trader to update their analysis and make decisions based on the latest information.

1.1-6

Describe a real-world problem in which sometimes the entire input is available

before you need to solve the problem, but other times the input is not entirely

available in advance and arrives over time.

Predicting the weather for the next week based on historical data and current conditions. Historical weather data, current temperature, humidity, pressure, wind speed, and other relevant meteorological factors.