



UGANDA CHRISTIAN UNIVERSITY

A Centre of Excellence in the Heart of Africa

DESIGN AND ANALYSIS OF ALGORITHMS ASSIGNMENT

NAME: MUKAMA JOSEPH

ACCESS NUMBER: B24267

REG NUMBER: S23B23/036

PROGRAM: BSCS

DESIGN AND ANALYSIS OF ALGORITHMS

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

- Users can find a certain contact fast by skimming through a contact list in a predictable order when the list is sorted alphabetically by last name. In daily applications, sorting is frequently used in this way.
- Network Routing: Data packets in computer networks are sent from their source to their destination via a network of linked nodes. By determining the shortest path between these nodes, network congestion is reduced and data delivery efficiency is guaranteed.

Other than speed, what other measures of efficiency might you need to consider in a real-world setting?

- Readability and Maintainability: How easy is it to understand and modify the algorithm? This can be important for long-term software projects.
- Robustness: How well does the algorithm handle unexpected inputs or errors? Fairness: In some applications, like machine learning, it's important to make sure that the algorithm is fair and doesn't exhibit biases.
- Space Complexity: What amount of space does the algorithm require to run? This can be particularly important in environments with limited memory, like mobile devices or embedded systems.
- Power Consumption: For battery-powered devices, algorithms that consume less energy can extend battery life.

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

Linked lists

strength:

- Effective insertion and deletion: The list is appropriate for dynamic data structures since elements can be added or removed at any point in time and at any position in the list.
- Dynamic size: Unlike arrays, which have a fixed size, linked lists have a dynamic size that can be changed as needed.
- Flexibility: Graphs, queues, stacks, and other data structures can all be implemented using linked lists.

Limitations:

- Inefficient random access: For big lists, it can be slow to have to go through the entire list in order to access an element at a certain index.
- Additional memory overhead: Compared to arrays, linked lists may use more memory because they need to retain pointers to the previous and next members.

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

Similarities:

- Both traveling salesperson and the shortest-path problem are optimization problems. In both cases, the objective is to choose the optimal option—in this example, the shortest path among a range of potential solutions.
- Graph-Based: Both issues can be represented as graphs, with nodes standing in for places and edges for the links that connect them.

Differences:

- Computational Complexity: The traveling salesperson problem is NP-hard, which means that no efficient solution exists for solving it precisely for large instances. In contrast, the shortest-path problem may be handled effectively using a variety of techniques.
- Applications: Network routing, navigation, and transportation planning are just a few of the broader uses for the shortest-path problem. More specialized applications of the traveling salesperson problem can be found in DNA sequencing, circuit board design, and logistics.

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.

Best solution

- Automatic Vehicles: In order to maintain safety and prevent collisions, self-driving cars must make precise decisions in real time. Algorithms that can precisely sense their surroundings, anticipate the actions of other cars and pedestrians, and make choices that reduce the chance of collisions are needed for this. A minor mistake in judgment could have far-reaching effects.

Approximate Best solution.

- Results from Search Engines: The goal of search engines is to give users the most appropriate details for their requests. Although precise outcomes are optimal, approximations that are however extremely important and beneficial can be adequate. Results that are almost perfect are usually acceptable to users as long as they are informative and meaningful.

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.

Weather Forecasting: To anticipate the weather in the future, weather forecasting models combine real-time observations with historical weather data.

- Entire Input Available: The model might have access to geographic data, climate models, and historical weather records.
- Over Time, Input Arrives: The model must update its forecasts based on real-time weather measurements from stations all over the world.