Engineering Method

Problem solving application.

Graph applications

Members:

- Alexis Jaramillo (A00395655)
- David Molta (A00372933)
- Alejandro Castro (A00372470)

Statement of the problem addressed: "Rappi is a delivery company that is actively seeking solutions to optimize its operations and reduce costs. One of its main challenges is the high expense associated with employee pay, which depends mainly on the route and distance traveled during deliveries. To solve this problem, Rappi intends to develop a program that calculates the minimum amount of money an employee should receive when moving from one location to another. This program will also provide the most efficient route for the employee to follow, ensuring that they reach their destination while minimizing costs.

In addition to its concerns regarding employee payments, Rappi also faces challenges with its internal mail system. Its network infrastructure consists of wires connecting multiple servers, each with its own specific latency measured in milliseconds. These latency values represent the time required to transmit a mail item between servers. To improve the efficiency of its mail delivery system, Rappi has decided to hire a software company specialized in solving this type of optimization problem.

The software company will develop a program capable of determining the shortest time required to send a message from a source server to a destination server. By identifying the most efficient wire sequence for transmitting mail, Rappi can significantly reduce the time it takes for messages to reach their recipients. Importantly, the program assumes that no delays occur on the servers themselves, focusing only on optimizing the transmission process.

Through these initiatives, Rappi aims to improve its overall operational efficiency, minimize costs associated with employee payments and improve the speed and reliability of its internal mail system."

Problem Context: The problem context of the statement is based on the challenges faced by the delivery company Rappi. The company seeks to optimize its operations and reduce costs by focusing on two main aspects:

1. Route calculation and employee payments - Rappi wants to calculate the minimum amount of money an employee should receive when moving from one location to another. This involves determining the most efficient route for the employee to follow, considering the distance traveled and minimizing the costs associated with employee payments.

2. Internal mail system optimization: Rappi has an internal mail system based on a network infrastructure with servers connected by wires. Each cable has a specific latency, which represents the time required to transmit mail between servers. Rappi seeks to reduce message delivery time by identifying the most efficient sequence of wires to transmit mail.

These issues pose challenges in terms of operational efficiency, costs associated with employee payments, and speed/reliability of the internal mail system. Our software company contracted by Rappi will develop a program capable of solving both problems, using at least two graph algorithms studied during the course, such as paths over graphs (BFS, DFS), minimum weight paths (Dijkstra, Floyd-Warshall) and minimum spanning tree (Prim, Kruskal). We will further specify the algorithms used to solve this presented problem.

By addressing these challenges, Rappi seeks to improve its operational efficiency, reduce costs associated with employee payments and streamline the internal mail delivery system, which will contribute to a better experience for both employees and the company's customers.

Solution development: To solve this problem, we will use a series of steps to follow, which will help us to have a systematic, effective, and efficient process to solve the need raised, which are:

1. Identification of the needs and symptoms of the problem: We can infer and inquire deeply into the problem that is presented to us, so we can highlight what are the aspects that are present in our case, we can lead to an effective solution:

To identify the needs and symptoms of the problem stated in the problem statement, it is important to analyze each of the two problems separately:

Problem 1: Calculation of routes and payments to employees.

Identified needs:

- Calculate the minimum amount of money an employee should receive when commuting from one location to another.
- Find the most efficient route to minimize costs associated with employee payments.
- Optimize Rappi's operations in terms of efficiency and costs.

Symptoms of the problem:

- High expense associated with employee pay, which depends on the route and distance traveled during deliveries.
- Lack of an automated system to calculate employee pay and find the most efficient routes.
- Possible inefficiencies in Rappi's current operations in terms of delivery costs and employee routing.

Problem 2: Optimization of the internal mailing system.

Identified needs:

- Reduce message delivery time in the internal mail system.
- Identify the most efficient wire sequence to transmit mail between servers.
- Improve the speed and reliability of Rappi's internal mail system.

Symptoms of the problem:

- Delays in message delivery in Rappi's internal mail system.
- Network infrastructure with servers interconnected by cables with different latencies.
- Lack of an optimized mechanism to determine the most efficient wire sequence and minimize message transmission time.

These identified symptoms and needs provide a clear picture of the challenges and opportunities Rappi faces in relation to employee routing and payments, as well as the efficiency and speed of the internal mail system. By addressing these issues, Rappi seeks to improve its operational efficiency, reduce costs, and offer a faster and more reliable service to both its employees and its customers.

2. Compilation of the information: For greater clarity in our search for a solution, we must have cleared all the elements that we are going to use, within which they can be:

- Graph: A graph is a data structure composed of a set of vertices (nodes) and a set of edges (connections) that link them. In our case, we will use graphs to represent the locations, paths, and connections between the elements of the problem.

- Vertex: Also known as a node, it is an individual element in a graph. In the context of the problem, vertices will represent locations, servers or other relevant points in the routing and mail transmission calculation.

- Edge: Also called a connection, it is a relationship between two vertices in a network. Edges will represent the routes, distances or latencies between locations or servers in the problem.

- Weight of an edge: A value associated with an edge that represents some measure, such as distance or latency, that is used in algorithms to find optimal routes or paths.

- Traversals over Graphs: These are algorithms that allow visiting the vertices of a graph in a systematic way. Two common types of tours are BFS (Breadth-First Search) and DFS (Depth-First Search). These algorithms are used to explore and analyze the structure of a network.

- Minimum Weight Paths: These are algorithms that find the shortest paths between two vertices in a graph. Two popular algorithms are Dijkstra and Floyd-Warshall. These algorithms are used to determine the most efficient path in terms of cost or distance.

- Minimum Covering Tree (MST): A subset of edges of a connected network that connects all vertices with the lowest possible weight. Two common algorithms for calculating the MST are Prim and Kruskal. These algorithms are used to optimize connectivity and reduce costs in problems such as route distribution or network infrastructure.

These concepts form the basis of graph theory and will allow us to address the problems posed in the statement by developing appropriate algorithms and data structures.

3. Search for creative solutions: Next, several solutions will be presented with which the problem can be developed, they were obtained through a brainstorming that resulted in the following options:

<u>Combined BFS and Dijkstra + Floyd-Warshall algorithm:</u>

- Use the BFS to explore all possible routes and find delivery locations.

- Apply Dijkstra's Algorithm to calculate the minimum amount of money an employee should receive when commuting and determine the most efficient route.

- Implement the Floyd-Warshall Algorithm to determine the shortest time required to send a message from a source server to a destination server in Rappi's internal mail system.

<u>Kruskal Algorithm + Dijkstra Algorithm:</u>

- Use Kruskal's Algorithm to find a minimum spanning tree (MST) that connects all delivery locations efficiently, minimizing costs associated with employee payments.

- Apply Dijkstra's Algorithm to calculate the minimum amount of money an employee should receive when commuting and determine the most efficient route.

<u>DFS Search Algorithm + Dijkstra's Algorithm:</u>

- Implement a depth-first search (DFS) algorithm to explore alternative routes and find more efficient delivery options.

- Use Dijkstra's Algorithm to calculate the minimum amount of money an employee should receive when commuting and determine the most efficient route.

<u>Dijkstra's Algorithm with time constraints + Floyd-Warshall Algorithm:</u>

- Extend Dijkstra's Algorithm to include time constraints and meet established deadlines.

- Implement the Floyd-Warshall Algorithm to determine the shortest time required to send a message from a source server to a destination server in Rappi's internal mail system.

Combined BFS and Dijkstra + Kruskal algorithm:

- Use the BFS to explore all possible routes and find delivery locations.
- Apply Dijkstra's Algorithm to calculate the minimum amount of money an employee should receive when traveling and determine the most efficient route.
- Use Kruskal's Algorithm to find a minimum spanning tree (MST) that connects all delivery locations efficiently.

DFS Search Algorithm + Kruskal Algorithm:

- Implement a depth-first search (DFS) algorithm to explore alternative routes and find more efficient delivery options.
- Use Kruskal's Algorithm to find a minimum spanning tree (MST) that connects all delivery locations efficiently, minimizing costs associated with employee payments.

These combinations of solutions address the problems of optimizing operations, reducing costs, and improving the efficiency of the internal mail system presented by Rappi.

4. Transition from Ideas to Preliminary Designs: To begin the selection of the best solutions to implement, it was decided to consider as evaluation criteria the development factors of efficiency, effectiveness, ease of implementation, support, and existing documentation. For this, it is necessary to analyze each of the proposed solutions. In the following, we will analyze each of the approaches:

Justification of the best 3 solutions:

Combined BFS and Dijkstra algorithm + Kruskal algorithm:

Rationale: This combination of solutions addresses both the routing and employee pay optimization problem and the problem of finding an efficient minimum overlay tree. The use of the BFS and Dijkstra ensures efficient and accurate routing for delivery drivers, while Kruskal's algorithm optimizes delivery routes in terms of distance and time. In addition, the implementation of these algorithms is extensively documented and resources and practical examples for their development are available.

DFS search algorithm + Dijkstra's algorithm:

Rationale: This combination provides an effective solution to the problem of finding alternative routes and optimizing employee pay. The DFS search algorithm allows exploring different paths and finding more efficient delivery options, while Dijkstra's algorithm calculates the minimum amount of money employees should receive when commuting and determines the most efficient route. These algorithms are widely used and have extensive and accessible documentation, making them easy to implement.

Dijkstra's algorithm with time constraints + Floyd-Warshall algorithm:

Rationale: This combination of solutions addresses both the employee payment optimization problem and the internal mail system efficiency problem. The extension of the Dijkstra algorithm with time constraints allows meeting the set delivery deadlines, while the Floyd-Warshall algorithm improves the efficiency of internal mail transmission. While the implementation of these algorithms may be a bit more complex than other options, there are resources and detailed documentation on how to apply and adapt them to this specific context.

The other options were not chosen due to the following reasons:

Combined BFS and Dijkstra algorithm + Floyd-Warshall algorithm:

This combination can provide accurate and efficient solutions, but by not addressing the employee pay optimization problem using an algorithm such as Kruskal, there may be a limitation in the overall effectiveness of the solution.

Combined BFS and Dijkstra + Kruskal algorithm:

While this combination is effective in addressing the routing and employee pay optimization problem, the addition of Kruskal's algorithm may increase implementation complexity, especially if a suitable data structure for the network representation is not available.

DFS search algorithm + Kruskal algorithm:

While this combination is effective for finding alternative routes and optimizing delivery routes, it does not address the problem of calculating the minimum amount of money an employee should receive when commuting. This could result in an incomplete solution to the given statement.

In summary, the three best solutions selected stand out for their effectiveness in fully solving the statement, their accuracy in optimizing paths and employee payments, their manageable level of implementation difficulty, the availability of adequate documentation, and their ability to address the key aspects of the problem posed by Rappi. From here we can submit these three solutions under evaluation criteria to select the best choice to implement and apply in our project.

5. Evaluation and Selection of the Best Solution: The 3 choices are compared under criteria to choose the best alternative solution.

The selected criteria are the following:

- Criterion A. Accuracy of the solution. The alternative delivers a solution:
  [2] Exact (an exact solution is preferred).
  [1] Approximate

- Criterion B. Efficiency. A solution with better efficiency than the others considered is preferred. The efficiency can be:
  [4] Constant
  [3] Greater than constant
  [2] Logarithmic
  [1] Linear

- Criterion C. Completeness. A solution that finds all solutions is preferred. How many solutions it delivers:
  [3] All
  [2] More than one if any, but not all.
  [1] Only one or none

- Criterion D. Ease of algorithmic implementation:
  [2] Compatible with the basic arithmetic operations of a modern computer hardware.
  [1] Not fully compatible with the basic arithmetic operations of a modern computer system.

Creative solutions	Criterion A: Accuracy	Criterion B: Efficiency	Criterion C: Complexity	Criterion D: Implementation	Total Points
Combined BFS and Dijkstra Algorithm + Kruskal Algorithm	2	2	3	2	9
DFS Search Algorithm + Dijkstra Algorithm	2	2	2	2	8
Dijkstra Algorithm with time constraints + Floyd-Warshall Algorithm	2	3	3	2	10

Under these criteria, both the combined BFS algorithm and Dijkstra + Kruskal's Algorithm and the DFS Search Algorithm + Dijkstra's Algorithm obtain an excellent overall score. Both solutions are accurate, efficient, and complete in terms of finding all necessary solutions. However, the solution selected as the best will be the Dijkstra Algorithm with time constraints + Floyd-Warshall Algorithm, as it stands out for its higher efficiency and ease of algorithmic implementation. In addition, the combination of these two algorithms directly addresses the problems posed in the Rappi statement and has greater flexibility by considering time constraints in the delivery of the target locations.