Report of Disaster Relief Management System Project

San San Maw & Ei Mon Soe

PARAMI University

DATA 301: Data Structures and Algorithms

Dr. Aye Hnin Khine

May 25, 2025

Problem Statement

Recently, a 7.7 magnitude earthquake hit Myanmar on March 28, 2025 (Tingle, 2025). According to the latest update as of 7 May, UNICEF reported that there are 3,791 deaths, 5,106 injuries, and 88 people are still missing (UNICEF, 2025). Even though the whole country was shocked by everyday challenges such as economic instability, loss of education, and loss of family members due to civil war since the coup, we Myanmar people are generous, and always eager to donate what they have to the poor or to the people who are in badly need of help. There are many social workers, donors, NGOs, and CSOs who are working on the ground, trying to meet the necessary needs for those who are affected by the earthquake. When to comes to donating necessary items to earthquake-affected people, our team "Pathfinders 7.7" wants to emphasize the system that can avoid being chaotic and being unorganized to make sure that the relief distribution goes systematically. Therefore, this project aims to create a command-line system that manages donation resources and distributes them to affected areas based on need and priority. This simple and organized system will make a real difference in facilitating donations and supporting local communities.

How Data Structures Power the Project

In the project, three data structures were implemented and tested as main functionalities. They are a singly linked list for tracking donations, a priority queue for relief area prioritization, and the Dijkstra Algorithm (Minimum Spanning Tree) to find the shortest path from a storehouse (starting point) to a disaster location. Those data structures are powerful and useful when optimizing disaster relief delivery.

- ***Singly Linked List***

The Singly Linked List is applied in this project to ensure high efficiency in terms of handling the donation resources. With the use of linked lists, the new donations can be easily added without requiring the resizing of the memory space. Each donation is stored in the node, and then, they are linked together so that it is flexible to add new donations to the list or remove them from the list. In the real-life scenario of having changes in the donation list record, a Linked List works well, and it supports organizing when we have to sort donations based on the location or urgency, if needed.

- ***Priority Queue***

After developing the Linked List for tracking donations, we used the priority queue as another important data structure. Based on the request for disaster aid by the urgency ranking 1, which is the highest priority, to 5, presenting the lowest priority, it supports the ideal situation where the critical places or events should be addressed first. Unlike a queue, a priority queue does not follow the order, but keeps the most urgent one at the front, whether the request arrives late or not.

- ***Dijkstra Algorithm (Shortest Path Algorithm)***

Dijkstra's Algorithm is applied afterwards as an additional data structure in order to search for the shortest path or way from the starting point to the different locations. The node represents the location, and the weight means the distance of the location or the time taken to get to the location. According to this, the closed unvisited location is picked, then the path is visited till all the nodes are visited. As an outcome, the dictionary of the minimum distance from the starting node (location) to other nodes is obtained, which is practically useful in the logistics aspect of distributing items to the locations affected by the disaster.

Main Functionalities And How It All Works

For main functionalities, there are 6 parts, which are add donation (Singly Linked List), remove donation (Singly Linked List), report disaster (Priority Queue), distribute supplies (Priority Queue/ Singly Linked List), view supplies (Linked List), and shortest path (Dijkstra Algorithm - MST). To add a donation and remove it, append () and delete () functions were used for the first part of the coding, respectively. After the steps are added and deleted, print_list() will print out all current donations stored in the linked list. As the second part of functionality coding, disaster reporting using a priority queue was applied. For that part, report_request () is the main mechanism when submitting a new aid request to the system. It uses a priority queue that ranks reported locations based on their urgency level. It is efficient when aid is sent to the most emergency areas. For the third part, it is distribute_supplies () function that is crucial for supplies logistics. It connects available donations to the most urgent aid requests. It will first check any pending aid requests and available donations. If both are found, it retrieves the next most urgent aid request from the priority queue. Then, it fetches the first available donation from the head of the linked list. After that, the function prints a confirmation that supplies from the donor are being sent to the affected location. After the donation is used, it updates the linked list by moving the head to the next node. After that, for the last part of functionality, it is dijkstra () function that works to find the shortest path from the starting point to the destination. The function analyzes each possible route and dynamically updates the minimum distance required to reach each location. Therefore, these six core functionalities form an integrated system that not only organizes donations but also prioritizes urgent needs and delivers supplies through the most efficient routes.

Complexity Analysis

Regarding time complexity analysis of the applied functions, the linked list as a donation list includes append and delete functions as key operations. They represent the time complexity of O(n), which means it needs to go through the entire list. As for the priority queue, report_request and get_next_request follow the time complexity of O(log n). This is an ideal system to prioritize the urgent requests. In order to get the shortest path by using Dijkstra's Algorithm, the time complexity of $O(n^2)$ is applied, which finds the shortest one by searching all unvisited nodes (locations) to find the shortest known distance. In this case, n represents the number of locations (or nodes). In each step, the algorithm selects the next closest unvisited location, which takes O(n) time. Since this process repeats for all n locations, the overall time becomes $O(n \times n) = O(n^2)$ for Dijkstra's Algorithm.

References

UNICEF. (2025, May 10). *UNICEF Myanmar Flash Update No. 12 (Earthquake), 09 May 2025 - Myanmar*. ReliefWeb. Retrieved May 24, 2025, from https://reliefweb.int/report/myanmar/unicef-myanmar-flash-update-no-12-earthquake-09-may-2025

Tingle, L. (2025, March 31). *How an earthquake in Myanmar also killed people and shook buildings in Bangkok*. ABC News. Retrieved April 26, 2025, from https://www.abc.net.au/news/2025-03-31/what-we-know-about-the-earthquake-in-myanmar/105116378