# Trekking Route Planning for Nepali Mountain Trails based on Time, Distance and Difficulty

Rohit Kumar Sah, Dipesh Duwal

## Big Idea

Nepal, a country known for its stunning mountain landscapes, has numerous trekking trails with their own challenges. However, finding the optimal trekking route while minimizing difficulty and balancing distance and available time remains a significant challenge for trekkers. Our idea is to develop a route planning system that suggests the easiest and the most efficient route from starting point to destination. This system will maximize the number of possible key locations while ensuring the trek remains feasible within the allotted number of days.

## Methodology

This problem can also be classified as a classic knapsack problem where we are trying to maximize the possible locations while minimizing the difficulty of the path. A high level plan on how we are going to approach the problem:

1. Greedy Algorithm Approach

   This approach will always make the optimal choice from the starting point, based on weighted scoring of attractions, distance, and difficulty. In this approach we will always choose the nearest unvisited location that can be reached within the remaining days. It is expected to be fast, but the result may turn out to be suboptimal. It will be tested with experiment 1 and 2 for scalability with large networks.

2. Divide and Conquer Approach

   In this approach we are going to break the trail into smaller subproblems recursively to find the best segments. Then the selected segments would be combined based on attractions coverage and difficulty. This approach is expected to be less memory efficient and suboptimal. Similarly, both experiment 1 and 2 will be tested using this algorithm.

3. Dynamic Programming Approach (0/1 Knapsack)

   In this approach we are going to create a dynamic programming table considering all possible combinations of places while maximizing total attractions coverage and minimizing difficulty while also accounting for the number of remaining days. This solution is going to be optimal but it will be the most computationally expensive one.

## Dataset

The project uses two primary datasets: (1) GeoJSON data having coordinates of major trekking trails of Gaurishankar Conservation Area, sourced from the National Trust for Nature Conservation, and (2) additional data from OpenStreetMap including altitude, location names, and attractions based on the trails coordinates. The initial dataset lacks critical information about trail locations and terrain difficulty. This data will be transformed into a weighted graph structure where nodes represent locations and edges contain properties like difficulty, distance, and nearby attractions. For experiments we will be generating nodes and edges randomly based on the requirements.

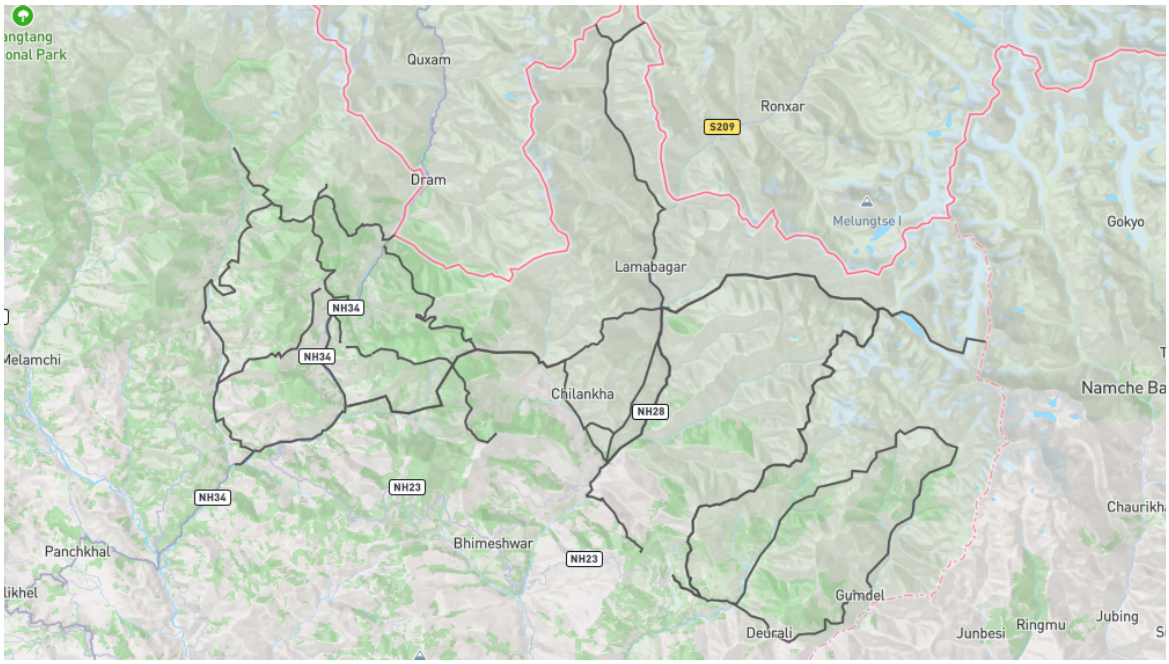


*Figure 1: A map view plot of the trekking trails from Gaurishankar Conservation Area based on trail coordinates.*

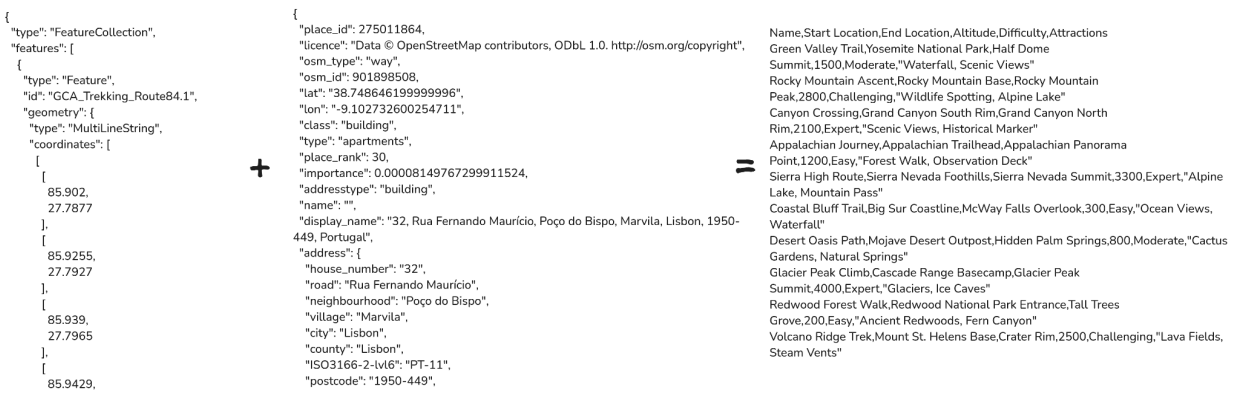


*Figure 2: Image showing the sample formats of the trail data (left), OSM data (middle) and expected data (right) after processing.*

**Experiments**

For all the experiments listed below, we'll be measuring runtime in seconds, memory usage in MB, and solution quality (using other metrics like overall difficulty and attractions coverage).

Experiment 1 evaluates algorithm scalability and solution quality across varying network sizes. The experiment will test generated networks of 10, 50, 100, 500, and 1000 vertices while maintaining a fixed 7-day limit and consistent difficulty distribution.

Experiment 2 examines how well our algorithms handle varying constraints, testing their robustness and adaptability. We'll vary day limits (3, 7, 14, 30 days), daily distance limits (10km, 20km, 30km), and difficulty thresholds. By measuring constraint satisfaction rates, path completion rates, and attraction coverage, we can analyze how frequently constraints are violated, what percentage of solutions remain feasible under different conditions, and how the algorithms balance competing constraints.

Experiment 3 examines how the algorithm performs on a real world dataset. Based on the start and destination point we can examine whether the solution routes match real world routes followed by trekkers. The experiment measures how similar paths are to established routes, and compares attraction coverage against traditional paths.

## Gantt Chart

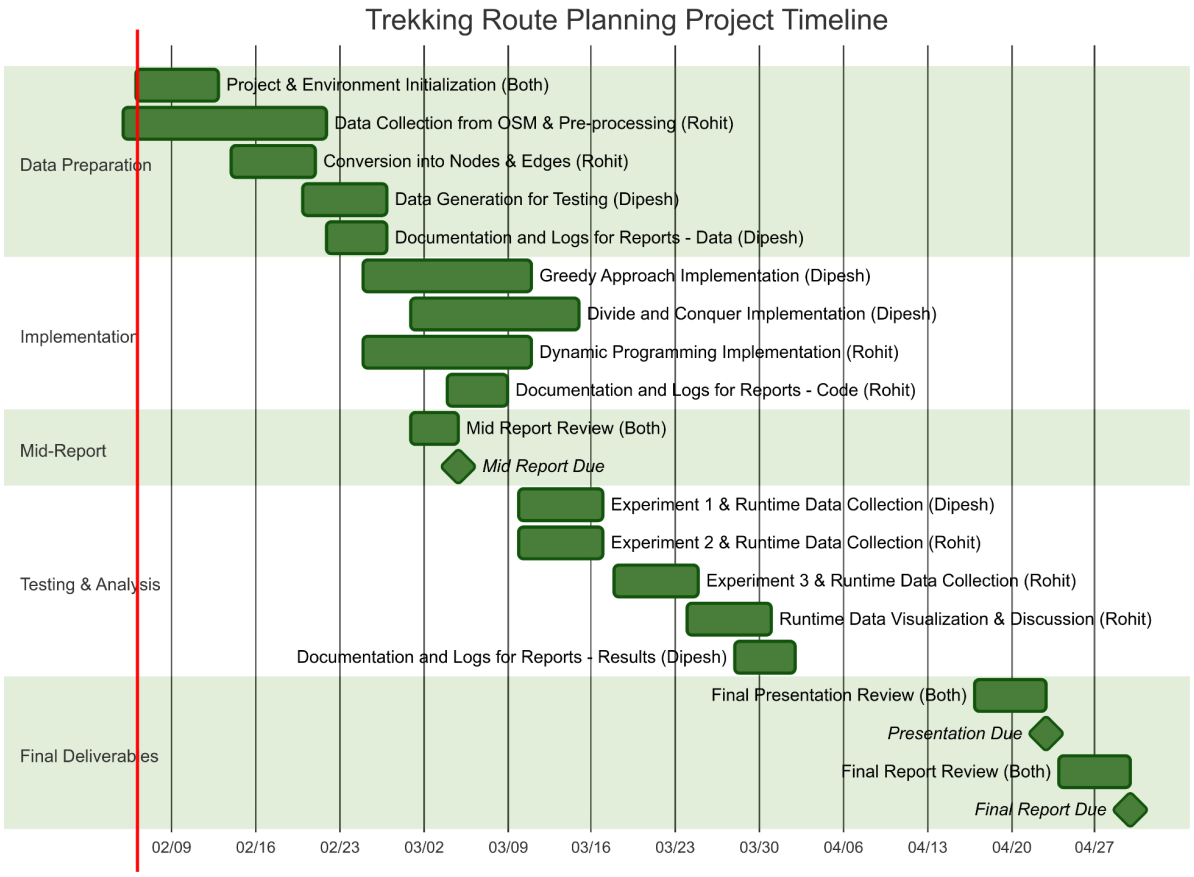


*Figure 3: Gantt chart showing timeline of the project.*