CS-214 Report

Design Choices

1. ADT- Graph

- Instances: line 77
- Role: The Graph ADT plays a crucial role in representing the connections between different locations in the trip planner. It allows us to model the relationships between various destinations and enables efficient path finding with use of its methods to find neighbors of one position and distances associated with each path.
- Data Structure: I implemented the WU_Graph ADT that uses an adjacency matrix representation.
- Reasoning: the adjacency matrix representation for the Graph ADT provides an efficient way to represent worst case, dense graphs. A dense graph's number of edges is large ($E \leq V^2$, worst case E is super close to V^2), an adjacency matrix can be more space-efficient as it requires $O(V^2)$ space, while an adjacency list may require $O(V + E)$ space, where V is the number of vertices and E is the number of edges.With an adjacency matrix, it is straightforward to determine whether an edge exists between two vertices; it provides constant-time access to the presence or absence of an edge.

2. ADT- PriorityQueue

- Instances: line 117, 165, 202
- Role: The PriorityQueue ADT is utilized for implementing Dijkstra's algorithm, which is employed to find the shortest path between two locations in find_nearby and plan_route, and for sorting when outputting results for find_nearby.
- Data Structure: I opted for a binary heap as the data structure for the PriorityQueue.
- Reasoning: A binary heap provides efficient insertion, deletion, and retrieval of the minimum element, which is essential for Dijkstra's algorithm. It guarantees a logarithmic time complexity for all these operations and this makes a binary heap suitable for dijkstra's which requires frequent updates to the priority queue. Binary heap is a data structure that can be implemented using other built-in data classes or structs, requiring only a small amount of additional memory compared to the elements being stored. This makes it memory-efficient, especially when dealing with large datasets.

3. ADT- Dictionary

- Instances: line 78, 79, 107, 167, 168,
- Role: The Dictionary ADT is used for storing key-value pairs, representing information related to points of interest, positions, and other entities in the trip planner.

- Data Structure: I implemented the Dictionary ADT using association lists.
- Reasoning: Association lists can easily grow or shrink dynamically as elements are added or removed. When a new key-value pair is added, it is simply appended to the existing list; this dynamic resizing capability makes an association list flexible in handling changes to the data structure, without requiring additional operations like resizing or rehashing as in hash tables or having any unused allocated memory.

4. Algorithm: Dijkstra's

- Lines: 133-146, 186-201
- Role: utilized to find the shortest path from a single source vertex to all other vertices in the graph.
- Reasoning: Dijkstra's algorithm is specifically designed to handle weighted graphs, where each edge has a non-negative weight. It efficiently calculates the shortest path by considering the weights of the edges between points and also flags unreachable points of interest by not changing their initially set distances of infinity. By maintaining a vector that keeps track of the previous vertex for each visited vertex, dijkstra's is straightforward to trace back the path from the destination vertex to the source vertex.

Open Ended Questions

1. Make businesses with discounts rank higher in find_nearby

- I would maintain two binary heaps: one for businesses with discounts and another for businesses without discounts. Insert businesses with discounts and the ones without discounts into the binary heap for businesses with discounts and the binary heap for businesses without discounts, respectively, considering the minimum distance priority. Once both binary heaps are populated, begin by extracting businesses from the binary heap for businesses with discounts until it is empty and if the number of businesses extracted from the heap with discounts is less than n, continue extracting businesses from the binary heap for businesses without discounts. By prioritizing businesses that offer discounts, users of the trip planner will have a higher chance of finding attractive deals and this promotes a sense of value and incentivizes cost-conscious decision-making, allowing users to make the most of their budget. Businesses may be more inclined to offer exclusive deals and promotions to the trip planner service, knowing that their offerings will receive preferential treatment in the search results. This can lead to mutually beneficial relationships and business growth. However, this could lead to a biased representation of businesses in the results, potentially overshadowing businesses that provide high-quality products or services but may not offer discounts, also users may be presented with a skewed selection of businesses that offer discounts, potentially limiting their exposure to different types of experiences or services. If find_nearby consistently prioritizes businesses with discounts, it may incentivize

businesses to offer deeper discounts or engage in price wars to maintain or improve their ranking. This could lead to lower profitability for businesses in the long run. I would not go on to implement this change exactly the way it's proposed because by prioritizing businesses with discounts, there is a risk of creating an unequal playing field for businesses that do not offer discounts. This can be perceived as unfair by those businesses and their customers. Implementing a ranking system solely based on discounts may undermine the principle of equal opportunity and fair competition among businesses, but I would approach my boss and suggest we have a toggle for customers to choose if they want relevance or discount-ranking.

2. Shadow ban businesses without any Alert

- The shadow ban feature can potentially improve the overall user experience by reducing the visibility of businesses that fail to meet certain criteria or standards, the feature can help maintain a higher quality of listings on the trip planner. This can promote reliable and reputable businesses, ensuring that users have access to more trustworthy options. However, implementing a shadow ban feature may raise concerns about transparency and fairness. Users may question the credibility of the platform if they suspect that certain businesses are being hidden without their knowledge. It can erode trust and lead to a negative perception of the trip planner's integrity. The decision to shadow ban businesses can be subjective and prone to bias, depending on the criteria and judgments applied by the boss, there is a risk of penalizing businesses unfairly or targeting them based on personal opinions, which can create controversies and legal issues. Shadow banning may have unintended consequences for businesses, such as a loss of visibility, reduced customer traffic, and financial impacts. It can disproportionately affect smaller or lesser-known businesses that heavily rely on the trip planner for exposure and growth. I will not agree to do it, because not allowing the business to see that they have been banned sounds suspicious and also because a shadow ban should serve as a means to provide feedback to businesses allowing them to assess their performance and make necessary improvements based on user feedback and reviews, potentially leading to better service quality in the long run and thus I believe a report as to why they were banned should be sent out and not hidden from the business.