**Data Structures and Algorithms**

**Project Evaluation Sheet**

Name: Nagendra Hanchinale SRN: 01FE22BCS041

Roll:206

Implementation Analysis

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Algorithm/Data Structure** | **Used? (Yes/No)** | **How and where?** | **Space Efficiency** | **Time Efficiency** |
| Arrays | Yes | Used for various data structures such as 'City', 'places', 'route', etc. | O(n) | O(1) for access, O(n) for insertion |
| Structures | Yes | Structs like 'City', 'places' are used to organize and store related data. | O(size of elements) | O(1) for accessing members |
| List | No |  |  |  |
| Stack | Yes | Used for DFS traversal in the 'dfs' function. | O(n) | O(1) for push/pop |
| Queue | Yes | Used for BFS traversal in the 'bfs' function. | O(n) | O(1) for enqueue/dequeue |
| Binary Tree | No |  |  |  |
| Binary Search Tree | Yes | Print the places in inorder. | O(n) | O(log n) for searching, insertion, deletion |
| AVL Tree | No |  |  |  |
| 2-3 Tree | No |  |  |  |
| Red-Black Tree | No |  |  |  |
| Trie | No |  |  |  |
| Heap | No |  |  |  |
| Lookup Table | No |  |  |  |
| Sparse Table | No |  |  |  |
| Fenwick Tree | No |  |  |  |
| Segment Tree | No |  |  |  |
| Skip List | No |  |  |  |
| Union-Find | No |  |  |  |
| Hashing | Yes | Used in Robin Karp | O(size of string) | Average-case O(1), worst-case O(n) |
| DFS | Yes | To print the spanning root | O(V + E) | O(V + E) |
| BFS | Yes | To find the neighbourhood hangout places | O(V + E) | O(V + E) |
| Bubble Sort | No |  |  |  |
| Selection Sort | No |  |  |  |
| Insertion Sort | No |  |  |  |
| Quick Sort | Yes | To sort the routed with respect to diatance. | O(n log n) average, O(n^2) worst case | O(log n) average, O(n) worst case |
| Merge Sort | Yes | To sort the places with respect to ratting. | O(n log n) | O(n) |
| Brute Force String Search | Yes | Used while displaying the places according asked ratting. | O(mn) | O(1) |
| Rabin Karp | Yes | Used is valid function to see whether the city\_name entered is valid or not. | O(n + m) average | O(1) |
| Boyer-Moore | No |  |  |  |
| Knuth-Morris-Pratt | No |  |  |  |
| Heap Sort | No |  |  |  |
| Kruskal | No |  |  |  |
| Prim | No |  |  |  |
| Dijkstra | Yes | To display the efficient path from source to destination. | O(V^2) | O(V^2) |
| Floyd | Yes | To find nearest city | O(V^3) | O(V^3) |
| Warshall | Yes | To see the directly connected cities. | O(V^3) | O(V^2) |
| Bellman-Ford | No |  |  |  |
| Any Other | No |  |  |  |

Other Analysis:

Number of Lines of Code Written: 2000

Number of Functions: 17

Design Techniques and Principles used:

* DFS
* BFS
* Brute Force String Search
* Rabin karp
* Quick Sort ,
* Merge Sort
* Dijkstra, Warshall
* Binary Search Tree
* Rolling Hashing