

COEN 4650-Flight Assistant Project Proposal Description

Qian Cheng and Zhou Shen

Abstract—Use Hash table [1] and a greedy algorithm [2] to help travelers find the best scheme to the destination before purchase the ticket, also show them the detailed information about their flight after purchase. With our Flight Assistant, travelers can save money and time on purchasing tickets; what's more, travelers can know detailed information about their flight clearly and on time once needed.

Index Terms— Hash table greedy algorithm

I. INTRODUCTION

Today more and more people select airplane as their transportation for work, travelling and meeting friends. And they sometimes have troubling in buying tickets because they don't know which tickets can saving more money and time [3]. And when they buy ticket long time before the flight time, someone may forget flight time and want a software so that they can search their flight time quickly [4]. Designing a software that help people solve these problem is important for it can save people much time and money in selecting and searching tickets. We will use Hash table for ticket searching problem and greedy algorithm that help people make choices in buying tickets.

II. PROPOSED APPROACH

1) Used Algorithm

(a) Hash Table

Hash Table, as a data structure which implements an associative array abstract data type, a structure that can map keys to values, no matter users want to type the departure and arrive places or find the detailed information about the flight by inputting their flight number, we can use Hash Table to map those input keys to the information users want. In details, the inputs will be what users type in for places or the flight number; Hash Table will transfer them into binary numbers then our project can understand; the outputs will be full name of airport and detailed information of selected flight.

(b) Greedy Algorithm

Greedy Algorithm, as the name suggests, always makes the choice that seems to be the best at that moment, This means that it makes a locally-optimal choice in the hope that this choice will lead to a globally-optimal solution. For this

problem, we want to find a best flight airline when we want to transfer. So we can divide this problem into some more concrete subproblems that the minimum time and minimum money that we will spend. The software has two objective functions, one is finding the minimum time of the flight, the other is finding the minimum money we will spend. The data contains the information about the flight between 10 cities. And the output is the minimum time or minimum money flight between two cities.

2) Inputs and Outputs

(a) Hash Table

Input: Brief airport code and flight number

Output: Full name of airport and detailed information about the flight

(b) Greedy Algorithm

Input: the time and cost about the flights between 10 cities.

Output: the minimum time or minimum money flights between two cities.

3) Programming Language

The language we will use is Java

And the data and information we will use from Google flights and the Google flights API.

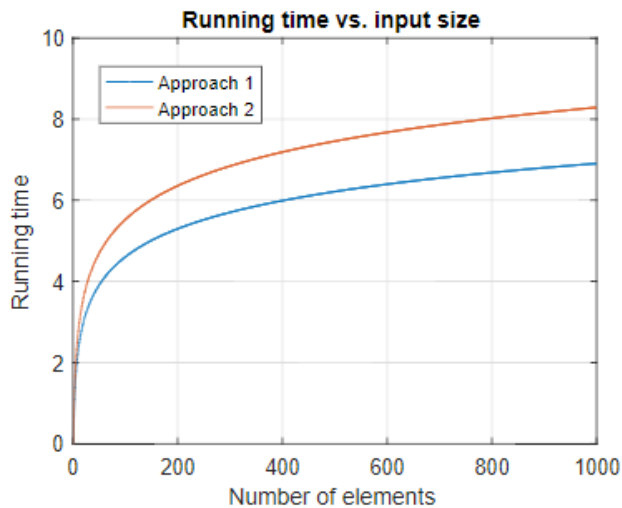
III. PROJECT TIMELINE AND MILESTONES

task ^T	excepted conclusion ^T	Milestones ^T
Study and understand Google flights library ^T	02/15 ^T	^T
Install and configure Google flights library ^T	02/25 ^T	^T
Implement I/O functionalities ^T	03/01 ^T	I/O module ^T
implement and evaluate correctness and performance of subroutine detecting ^T	03/07 ^T	subroutine detecting test module ^T
implement and evaluate correctness and performance of subroutine optimizing ^T	03/15 ^T	subroutine optimizing test module ^T
Implement, test, and evaluate project prototype ^T	03/28 ^T	Project Prototype ^T
Integrate subroutines detecting and optimizing and evaluate performance and correctness of integrated module ^T	04/15 ^T	integrated test module ^T
Implement main module ^T	04/20 ^T	^T
Evaluate the correctness and performance of the main module ^T	4/25 ^T	Main module ^T
Compile results and prepare project poster ^T	5/2 ^T	Project poster ^T

IV. EVALUATION PLAN

In order to evaluate the correctness of our project, we plan to have two tests for our two algorithms.

First, for Hash table, we are trying to figure out two different methods on the running time with different numbers of flights. Second, for greedy algorithm, we are attempting to use two different objective functions to get the minimum flight time. Then we will draw a figure about how the running time of two methods vary with different numbers of flights. Below is a sample of the figure.



Reference

1. "Hash functions.," *Princeton University*. [Online]. Available: <https://algs4.cs.princeton.edu/34hash/>. [Accessed: 04-Feb-2018].
2. "Basics of Greedy Algorithms Tutorials & Notes | Algorithms," *HackerEarth*. [Online]. Available: <https://www.hackerearth.com/practice/algorithms/greedy/basics-of-greedy-algorithms/tutorial/>. [Accessed: 04-Feb-2018].
3. "Airline booking ploys," *Wikipedia*, 22-Jan-2018. [Online]. Available: https://en.wikipedia.org/wiki/Airline_booking_ploys. [Accessed: 04-Feb-2018].
4. "FlightAware," *Wikipedia*, 28-Jan-2018. [Online]. Available: <https://en.wikipedia.org/wiki/FlightAware>. [Accessed: 04-Feb-2018].