Flight Assistant

Employing Hash Table store approach, quick sort algorithm and KMP search approach to finding the cheapest ticket from selected airports

Qian Cheng; Zhou Shen
Marquette University

Introduction

Today more and more people select airplanes as their transportation for work, travelling, meeting friends or doing business. However, sometimes they have trouble buying tickets because they don't know how to search for the airline effectively in order to save money.

Designing a software that helps people solve this problem is important because it can help people save a lot of money at most of time while selecting and searching tickets. We will use Hash table for storing airports and tickets information, quick sort algorithm to list all the airline ticket in decreasing order and KMP algorithm to find the cheapest ticket, showing users decreasing order list for the cheap ticket and cheapest ticket(s) between two selected airports.

Methods

Hash Table store approach. a hash table (hash map) is a data structure which implements an associative array abstract data type, a structure that can map keys to values. We first converted airport code to ASCII number and combined departure and arrivals number together as index; then we imported airline ticket information into table as key values.

Quick Sort algorithm. Quicksort (sometimes called partition-exchange sort) is an efficient sorting algorithm, serving as a systematic method for placing the elements of an array in order. We used quick sort algorithm to sort the whole ticket information between two selected airports and listed them in decreasing order based on price.

KMP search algorithm. The basic idea behind KMP's algorithm is: whenever we detect a mismatch (after some matches), we already know some of the characters in the text of next window. We used it to Search for Patterns. From previous list, we found the cheapest ticket and printed out with this algorithm.

Results

Analyzing the algorithm, the time complexity is O(1) + O(n*lg(n)) + O(n) which is O(n*lg(n)) running time for whole project.
Detailed information: running time of our Hash Table is O(1), of our quick sort algorithm is O(n*lg(n)) and of our KMP algorithm is O(n).

The graph below displays the behavior of our algorithm over time. The accuracy of this was tested by running the algorithm in the opposite direction, and verifying using other websites.

depature	arrivals	number of airline tickets	running time (ms)
MKE	ORD	6	20
	SEA	8	21
	SFO	5	18
	LAX	4	16
ORD	MKE	10	25
	SEA	8	22
	SFO	9	26
	LAX	13	30
SEA	MKE	4	17
	ORD	9	25
	SFO	8	23
	LAX	8	21
SFO	MKE	5	18
	ORD	11	27
	SEA	14	33
	LAX	12	27
LAX	MKE	5	17
	ORD	13	28
	SEA	12	26
	SFO	14	30

Table 1. number of airline tickets and running time table

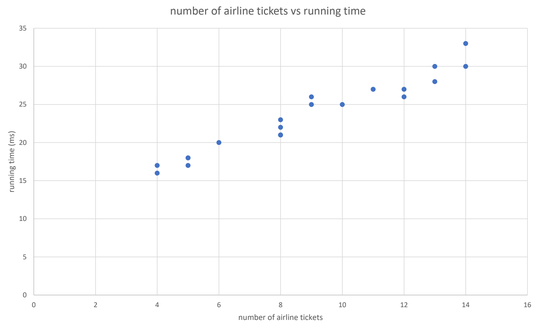

Figure 1. Number of airline tickets vs running time graph

```
Please choose one airport for departure
MKE
MKE
777569
Please choose one airport for destination
LAX
LAX
MKELAX
777569766588
```

Figure 2. Departure and arrival airports demo

```
777569766588 [08:25-11:01UA147,08:45-11:28UA188,12:07-15:35DA208,17:05-20:27DA214]
08:25-11:01UA147,08:45-11:28UA188,12:07-15:35DA208,17:05-20:27DA214
sorted result based on price
147
188
208
214
Information about our cheapest ticket:
08:25-11:01UA
```

Figure 3. Final results demo

Figure 4. Flow chart for our project

Conclusion

Our algorithm proved to be successful in storing the airline information into hash table and finding the cheapest airline. By using our flight assistant algorithm, people can easily find the cheapest airline information so that they can save money. The algorithm is convenient for searching the with a time-complexity of (nlgn).

References

1. https://en.wikipedia.org/wiki/Airline_booking_ploys.
2. https://en.wikipedia.org/wiki/Hash_table
3. https://en.wikipedia.org/wiki/Quicksort
4. https://en.wikipedia.org/wiki/Knuth%E2%80%93Morris%E2%80%93Pratt_algorithm
5. https://www.geeksforgeeks.org/searching-for-patterns-set-2-kmp-algorithm/