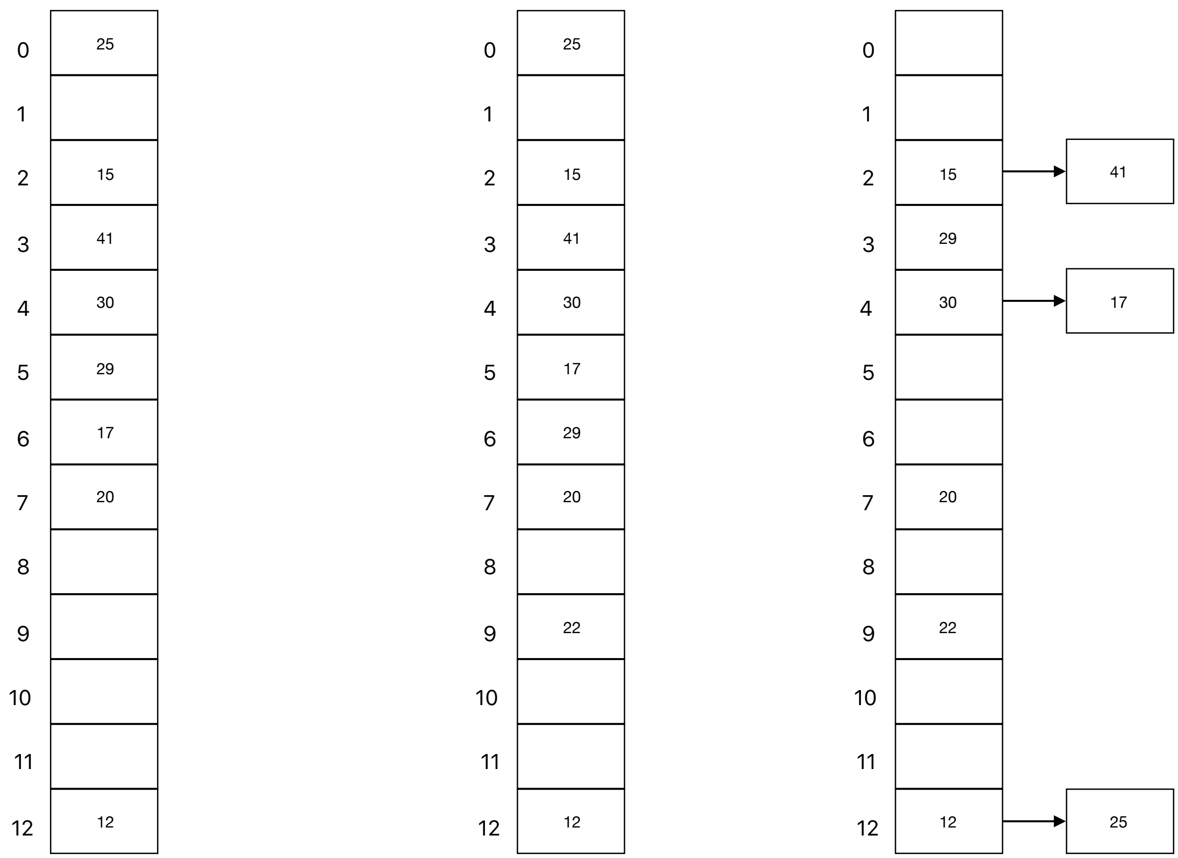
Title : Hashing and Graphs
Author : Bartu Atabek
ID : 21602229
Section : 2
Assignment : 4
Description : Contains answers to questions 1 & 3.

Question 1:

a) Insert {12, 15, 20, 30, 41, 29, 17, 25, 22} into an empty hash table.



Linear Probing Quadratic Probing Separate Chaining

b) Find the average number of probes for a successful search and an unsuccessful search for the hash tables which you created in part a. Use the following numbers for unsuccessful searches: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 38}

Table 1: Average number of probes

	Successful Search		Unsuccessful Search	
	Calculated	Theoretical	Calculated	Theoretical
Linear Probing	1.667	2.125	2.293	5.781
Quadratic Probing	1.778	1.703	3.154	3.250
Separate Chaining	1.333	1.346	0.692	0.692

Question 3:

In the part a of the second question if adjacency matrix used for the implementation of the graph the operation could be performed at constant time after finding the indexes in the array which would take O(n) time where n is the number of vertices. Since the function is only checking the value of an entry in the two dimensional matrix the overall worst case running time of the operation of part a is going to be O(n). On the other hand if the adjacency list would be used in order to represent the graph; the linked list which begins with the source airport has to be traversed. Because of this, the degree of the airport (number of edges that the airport has) is an important factor here. First of all the index of the source airport has to be located, which is in worst case O(n) and then the following linked list needs to be traversed which is going to take O(D) time.

In part c of the second question, the function requests the existence of a path between two airports which requires the traversal of the paths. For the worst case running times of this algorithm, there are two possible cases that could occur. First is that the path from A to B could require traversing all the vertices, secondly the path could be the last path which will require the program to go through all the edges. Within these observations the worst case running time of the function at part c would be O(|E| + |V|), where |E| represents the number of edges and |V| represents the number of vertices.

In part d, it is ask that the path between two airports A and B with the minimum number of stops to be outputted. According to the breadth first traversal algorithm, in the worst case the path between these two airports would require the whole graph to be traversed, which results with the same time complexity as in the part c which is O(|E| + |V|), where |E| represents the number of edges and |V| represents the number of vertices.