**Data Structures and Algorithms**

**Assignment 3**

**Jan 1 Semester**

**Name: Jeslyn Ho Ka Yan**

**ID: 8535383**

**Date of Submission: 1 March 2024**

**Table of Content**

1 Question 1 3

1.1 a) When Separate Chaining is used to handle collisions 3

1.2 b) When Collision are handled by linear probing 3

1.3 C) When collison are handled by double hashing 3

1.4 d) When Collision are handled by quadratic probing 4

2 Question 2 5

2.1 a) Insert element into partially constructed AVL 5

2.2 b) Make the array into a Maximum Heap 5

3 Question 3 7

4 Question 4 8

4.1 Adjacency Matrix 8

4.2 Show all paths from 0 to all other network nodes. 8

5 Question 5 11

5.1 Is the heuristic in the problem admissible? 11

5.2 Find the shortest path using A\*Search 12

# Question 1

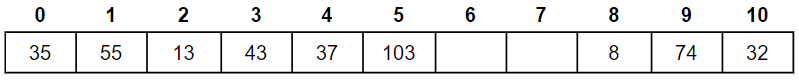
**Hash Function:**

**Value:**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **The hash** |  | 33 | 13 | 32 | 8 | 55 | 37 | 43 | 74 | 103 |
| **address:** |  | 0 | 2 | 10 | 8 | 0 | 4 | 10 | 8 | 4 |

## A number and arrows pointing to a number Description automatically generated with medium confidenceA close up of a sign Description automatically generateda) When Separate Chaining is used to handle collisions

## b) When Collision are handled by linear probing

A close up of black text

Description automatically generated

## C) When collison are handled by double hashing

**Second Hash Function:**

**Overall Function**:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **The hash** |  | 33 | 13 | 32 | 8 | 55 | 37 | 43 | 74 | 103 |
| **address:** |  | 0 | 2 | 10 | 8 | 0 | 4 | 10 | 8 | 4 |
|  |  |  |  |  |  | 1 |  | 4 | 5 | 4 |

Since , which is already occupied, collision occurs.

To solve this problem, we use the second hash function is applied.

, 1 is the new location and its not occupied, the vaule will be place in location 1.

Similar for .

However as you can see there is is still collision between .

In order to solve this, we will then use the overall function.

**For ,**

Let i be 0, (location 10 is occupied, collision)

Let i be 1, (location 3 is empty, no collision)

(part c will be continiued in the next page)

**For ,**

Let i be 0, (location 4 is occupied, collision)

Let i be 1, (location 8 is occupied, collision)

Let i be 2, (location 1 is occupied, collision)

Let i be 3, (location 5 is empty, no collision)

**Hash table using Double-hashing Resolution**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| 33 | 55 | 13 | 43 | 37 | 103 |  | 74 | 8 |  | 32 |

## d) When Collision are handled by quadratic probing

**Quadratic probing Function:** 2

Collsion had occur for . We have to use the Quadratic probing Function to handle the collision.

**For ,**

Let i be 1, 2)

(location 1 is empty, no collision)

**For ,**

Let i be 1, 2)

(location 0 is occupied, collision)

Let i be 2, 2)

(location 1 is empty, no collision)

**For ,**

Let i be 1, 2)

(location 9 is empty, no collision)

**For ,**

Let i be 1, 2)

(location 5 is empty, no collision)

**Hash table using Quadratic Probing Resolution**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| 33 | 55 | 13 | 43 | 37 | 103 |  |  | 8 | 74 | 32 |

# Question 2

## a) Insert element into partially constructed AVL

A diagram of steps and numbers

Description automatically generated with medium confidence Insert value: 7, 8, 3, 9

## b) Make the array into a Maximum Heap

A diagram of a diagram

Description automatically generated Value: 65, 57, 85, 48, 47, 61, 75, 5, 25, 85, 20, 42

A diagram of a diagram

Description automatically generated

A diagram of a diagram

Description automatically generated

A white background with black dots

Description automatically generated

(continue at next page)

# Question 3

* Quicksort with middle-of tree (mean) pivot
* Sorting the array in a accending order.

A screenshot of a table

Description automatically generated

A screenshot of a diagram

Description automatically generated

A diagram with numbers and a chart

Description automatically generated with medium confidence

# Question 4

## Adjacency Matrix

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **A** | **0** | **1** | **2** | **3** | **4** | **5** |
| **0** | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| **1** | 5 | ∞ | ∞ | ∞ | ∞ | 3 |
| **2** | ∞ | 6 | ∞ | ∞ | ∞ | 10 |
| **3** | ∞ | ∞ | 2 | ∞ | 2 | ∞ |
| **4** | ∞ | ∞ | 3 | ∞ | ∞ | 9 |
| **5** | 4 | ∞ | ∞ | ∞ | ∞ | ∞ |

## Show all paths from 0 to all other network nodes.

**Method 1(both answer are the same)**

A diagram of a diagram

Description automatically generated

A diagram of steps and answer

Description automatically generated

**Method 2 (both answer are the same)**

**A table with numbers and letters

Description automatically generated**

A white sheet with numbers and purple lines

Description automatically generated with medium confidence

A white grid with black numbers

Description automatically generated

A white rectangular grid with black numbers

Description automatically generated

A table with numbers and symbols

Description automatically generated

A screenshot of a number grid

Description automatically generated

A white grid with black numbers

Description automatically generated

A blue rectangle with black text

Description automatically generated

# Question 5

## Is the heuristic in the problem admissible?

For every node n, ∀(𝑠,𝑠′) :ℎ(𝑠)−ℎ(𝑠′) ≤ 𝐶(𝑠,𝑠′).

|  |  |  |  |
| --- | --- | --- | --- |
| **States** |  |  |  |
| **S-C** | 30 – 27 = 3 | C(S,C)= 3 | True |
| **S-B** | 30 – 17 = 13 | C(S,B)= 14 | True |
| **S-G** | 30 – 0 = 30 | C(S,G)= 33 | True |
| **C-D** | 27- 18 = 9 | C(C, D)= 9 | True |
| **B-E** | 17 – 18 = 7 | C(B, E)= 8 | True |
| **B-F** | 17 – 10 = 14 | C(B, F)= 15 | True |
| **D-F** | 18 – 3 = 15 | C(D, F)= 16 | True |
| **D-E** | 18 – 3 = 8 | C(D, E)= 8 | True |
| **E-F** | 10 – 3 = 7 | C(E, F)= 7 | True |
| **F-G** | 3 – 0 = 3 | C(F, G)= 4 | True |

Answer: Yes, the heuristic specified in the problem is admissible.

## Find the shortest path using A\*Search

|  |  |  |
| --- | --- | --- |
|  |  |  |
| S-C | S-B | S-G  Red text on a white background  Description automatically generated |
| SC-D | SB-F  Red text on a white background  Description automatically generated | SB-E  A close up of red text  Description automatically generated |



|  |  |
| --- | --- |
| SCD-F  A close up of red text  Description automatically generated | SCD-E |
|  | SCDE-F |
|  | **SCDEF-G** |

Answer: There is no need to continue as the goal is G. The shortest route is SCDEFG at cost 31.