## Q1 Linear probing

20 Points

The following keys are inserted into an initially empty linear-probing hash table.

| key | hash |
|---|---|
| 4 | 5 |
| 6 | 0 |
| 11 | 5 |
| 14 | 1 |
| 16 | 3 |
| 18 | 5 |
| 24 | 4 |

Suppose that the size of the hash table is fixed as 7 and the insertions are not necessarily in the order given above. Please specify which one or more of the following could be the contents of the resulting array? **You must mark all required answers to get points.**

- ☐ 24 11 14 4 18 16 6
- ☑ 6 14 18 16 24 11 4
- ☐ 4 6 14 16 24 18 11
- ☐ 24 14 11 6 16 4 18
- ☐ 18 14 4 6 16 11 24

## Q2 Separate chaining

20 Points

Suppose that the following keys are inserted in the given order into a separate-chaining hash table with 3 chains.

| key | hash |
|---|---|
| I | 1 |

| | |
|---|---|
| B | 0 |
| G | 2 |
| C | 1 |
| V | 2 |
| O | 1 |
| X | 1 |
| M | 2 |
| F | 1 |
| W | 0 |
| Y | 2 |
| L | 1 |

Consider one searches for the key G. Please specify the sequence of keys that are compared during this search. **Your answer should be a sequence of letters**
**separated by whitespace, e.g. I B G.**

## **Q3** Size of hash table

20 Points

Suppose that 600 items will be stored in a hash table and the average number of accesses in a successful search is expected to be about 2.0.

Load factor $\alpha$ is defined as the ratio of $N/M$ that is the average number of elements in the hash table.

### Q3.1

10 Points

What is the size ($M$) of the hash table for linear probing?
NOTE: For a load factor of $\alpha$, the average number of accesses is generally $\frac{1}{2}(1 + \frac{1}{(1-\alpha)})$.

3/2

### Q3.2

10 Points

What is the size ($M$) of the hash table for separate chaining?
NOTE: For a load factor of $\alpha$, the average number of accesses is generally $(1 + \frac{\alpha}{2})$

3/2

## **Q4** Sparse matrix

10 Points

Which of the following is **not** a suitable data structure to represent a Sparse Matrix?

○ Array

● Heap

○ Linked List

○ Dictionary of Keys (Symbol Table)

## **Q5** Hashcode and equals

20 Points

A programmer defines a class called $MyNumber$ but has no clue about what $hashCode$, $equals$ and $compareTo$ functions does. Answer the following questions for the code snippet below;

```
HashST<MyNumber, Double> map = new HashST<MyNumber, Double>();
MyNumber num = null;
for(int i=0; i<=100000; i++) {
   num = new MyNumber(i);
   map.put(num, i);
}
```

### **Q5.1**

6 Points

If the programmer does **not** implement any of $hashCode$ or $equals$ which of the following is/are true.

☑ $map.get(newMyNumber(3))$ returns $null$

☑ $map.get(num)$ returns a double equal to $100000$

□ Average case complexity of get is $3-5$ (i.e. $O(1)$)

## Q5.2

6 Points

If the programmer implements only the equals function (correctly by comparing to the value of MyNumber) which of the following is/are true.

☑ $map.get(newMyNumber(i))$ returns $null$ for some $i$ but returns a double with value $i$ for the others

☑ Average case complexity of get becomes $O(N)$

□ $map.get(num)$ returns a double equal to $100000$ (for sure)

## Q5.3

8 Points

If the programmer implements equals function (correctly by comparing to the value of MyNumber) but implements hashCode by just returning a constant value (e.g. 17) which of the following is/are true.

□ $map.get(newMyNumber(3))$ returns $null$

□ $map.get(num)$ returns null

□ Average case complexity of get becomes $O(N)$

## Q6

10 Points

Which of the following comparisons are correct for red-black trees and hash tables when used for the same set of keys.

☑ Hash tables use more space in comparison to Red-black trees.

□ Hash tables can find the rank of a key faster than Red-black trees.

☑ Red black tree provides a complexity guarantee of $O(logN)$ while hash tables has worst case of $O(N)$

☑ For both Red-black trees and the hash tables we need to implement the same functions (hashCode, compareTo and equals) in Java.

# Quiz 1 - Hashing

● GRADED

STUDENT

Mehmet Taha Usta

TOTAL POINTS

**28.5 / 100 pts**

QUESTION 1

Linear probing **0** / 20 pts

QUESTION 2

Separate chaining **0** / 20 pts

QUESTION 3

Size of hash table **0** / 20 pts

3.1 (no title) **0** / 10 pts

| | | |
|---|---|---|
| 3.2 | (no title) | **0** / 10 pts |

QUESTION 4

| | |
|---|---|
| Sparse matrix | **10** / 10 pts |

QUESTION 5

| | | |
|---|---|---|
| Hashcode and equals | | **11** / 20 pts |
| 5.1 | (no title) | **4** / 6 pts |
| 5.2 | (no title) | **2** / 6 pts |
| 5.3 | (no title) | **5** / 8 pts |

QUESTION 6

| | |
|---|---|
| (no title) | **7.5** / 10 pts |