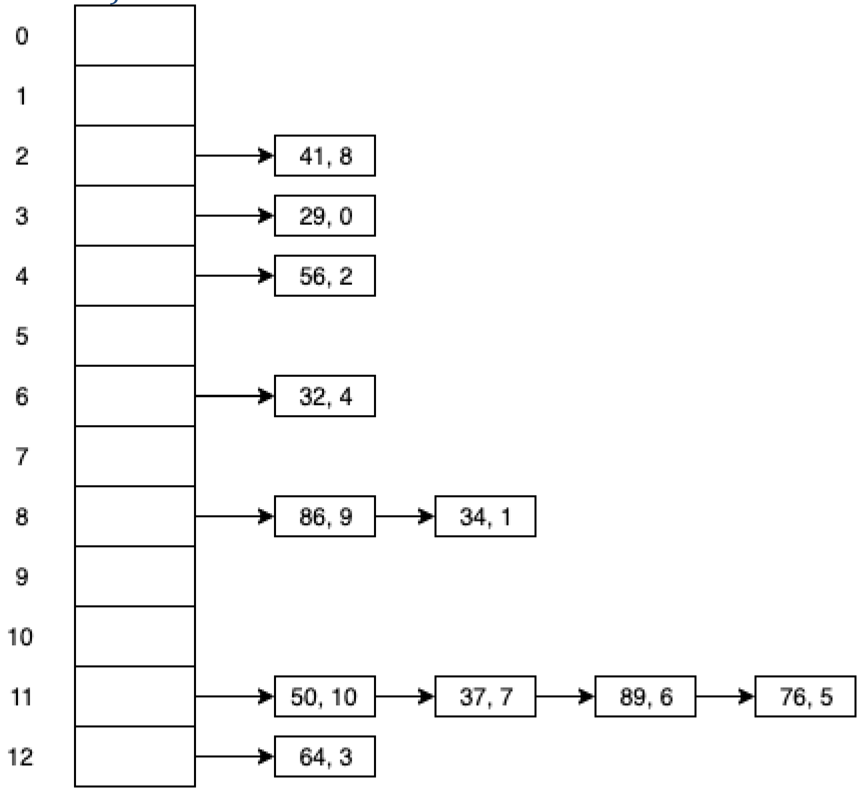
1. (10 Points) For B-Tree of order M discussed in class, answer the following questions.
a) For a non-leaf node (except root), what is the maximum keys it can have?
b) For a non-leaf node (except root), what is the minimum keys it must have?
c) For root node, what is the maximum keys it can have?
d) For root node, what is the minimum keys it must have?
e) Can leaf nodes have different depth?
Answer:
a) M-1
b) ceiling(M/2)-1
c) M-1
d) 1
e) No

2. (10 Points) Given keys as {29, 34, 56, 64, 32, 76, 89, 37, 41, 86, 50} and their associated values as {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10} respectively. Show the contents of the hash table after inserting those key values pairs into an initially empty hash table with size 13 using the following conflict resolution methods discussed in class:
a) Separate chaining. When there is a search miss, insert new key value pair at the beginning of the linked list.
b) Linear probing **with resizing**. Make sure to follow LinearProbingHashST code presented in class slides.

Answer: a)



b)

Index	0	1	2	3	4	5	6	7	8	9	10	11	12
keys				29	56		32		34	86		89	64
vals				0	2		4		1	9		6	3
Index	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**
keys	37		41									76	50
vals	7		8									5	10

3. (10 Points) Given a set with 13 elements, show the final result of executing the following instructions with UF_WeightedQuickUnion: union(9, 10), union(12, 1), union(1, 11), union(4, 9), union(11, 0), union(4, 6), union(3, 8), union(8, 4), union(3, 5), union(0, 8). Assuming initially there are 13 components.
a) Show the final contents of id[] array.
b) Draw the forest of trees after each union operation (in total 10 forests are required).
Answer:
a)

Index	0	1	2	3	4	5	6	7	8	9	10	11	12
id	12	12	2	9	9	9	9	7	3	9	9	12	9

b) Please checked assignment4-q3.png file.s

4. (10 Points) When using division method to map keys to table indexes, it is normally better to use a prime number for table size. Provide mapped indexes for the keys in the table using this division method: index = key %m.

Keys	200	2000	20,000	200,000	2000,000
m=100	0	0	0	0	0
m=101	99	81	2	20	99

5. (5 Points) Besides using division to map keys to table indexes, multiplication can be used as well. Use the following method to map keys to table indexes:
index = floor(m * ((key * A) mod 1)), where m=100, A = ($\sqrt{5}$ – 1)/2, and
((key * A) mod 1) means the fraction part of (key * A). For instance, key 100 will be mapped to index 80.

Keys	200	2000	20,000	200,000	2000,000
Mapped Indexes	60	6	67	79	97

6. (10 Points) Quadratic probing is a collision resolution method that eliminates the primary clustering problem of linear probing. It uses keys and vals arrays like linear probing. The difference between them lies in probe sequence. The probe sequence for quadratic probing is the following, where m is the table size
hash(key)%m,
(hash(key)+1*1)%m,
(hash(key)+2*2)%m,
(hash(key)+3*3)%m,
…
(hash(key)+i*i)%m, // for ith (zero based) probing
Provide the results of inserting following keys, 16, 8, 32, 9, 40, into the table using quadratic probing. If a key cannot be inserted into the table, indicate the key.
Table size (m) is 7

index	0	1	2	3	4	5	6
After inserting 16			16				
After inserting 8		8	16				
After inserting 32		8	16		32		
After inserting 9		8	16	9	32		
After inserting 40		8	16	9	32	40	

Table size (m) is 8

index	0	1	2	3	4	5	6	7
After inserting 16	16							
After inserting 8	16	8						
After inserting 32	16	8			32			
After inserting 9	16	8	9		32			
After inserting 40	16	8	9		32			

40 cannot be inserted into the table, because it keeps checking indexes 0, 1, 4, 1 repeatedly, which are occupied.

7. (5 Points) Double hashing uses a hash function of the form
hash(key, i) = (h1(key) + i*h2(key)) mod m, where i is the ith probe and is zero based, m is the table size, h1 and h2 are auxiliary hash functions.
Given m = 7, h1 = key % m, and h2 = 5 – (key % 5), provide the results of inserting following keys, 82, 41, 31, 75, 36, into the table.

index	0	1	2	3	4	5	6
After inserting 82						82	
After inserting 41						82	41
After inserting 31				31		82	41
After inserting 75		75		31		82	41
After inserting 36		75	36	31		82	41

8. (10 Points) Implement union method in the provided UF_WeightedQuickUnion class, which considers heights instead of sizes when joining components.

```
/**
* The implementation should be similar to weighted union using size, except this
* one uses height.  If heights of roots are different, the root with smaller
* height should be set as the child of the other root.  Otherwise, the second
* root should be set as the child of the first root.
 */
@Override
public void union(int p, int q) {
        // provide your implementation here
}
```

Answer: Here is one solution

```
public void union(int p, int q) {
    int pRoot = find(p);
    int qRoot = find(q);
    if (pRoot != qRoot) {
        if (height[pRoot] < height[qRoot]) {
            id[pRoot] = qRoot;
        } else  if (height[pRoot] > height[qRoot]) {
            id[qRoot] = pRoot;
        } else {
            id[qRoot] = pRoot;
            height[pRoot] += 1;
        }
        count--;
    }
}
```

Submission Note

1) For written part of the questions:
   a. Write your answers inside a text document (in plain text, MS Word, or PDF format)
   b. Name the file as csc130.firstname.lastname.assignment4.txt(doc, docx, or pdf) with proper file extension
2) For programming part of the questions
   a. Use JDK 1.8 and Junit5
   b. Put your full name at the beginning of every source file you created or modified. **2 points will be deducted if your names are not included in the source files.**
   c. **Do not change the provided package, class, or method name.** You can add extra classes or methods if they are needed.
   d. **If your code does not compile, you will get zero point**.
   e. Use the provided tests to verify your implementation. **Extra tests might be used for grading.**
   f. Zip all the source files into csc130.firstname.lastname.assignment4.zip
3) Submit both of your files (text document and zip file) via Canvas course web site.
4) Due Nov 22, 11:59 PM