

Write your name in box below and NetID to the right ==>	netid
Your name goes here	1pm Lecture <input checked="" type="checkbox"/>
	2pm Lecture <input type="checkbox"/>
	3pm Lecture <input type="checkbox"/>

I wish you all good skill! Read these instructions FIRST!

Please put your name and NetID neatly on this page and your name on the rest of the pages. I unstaple these to feed them into the scanner and I have (at least once!) dropped the stack of exams and had to put them back together...

Don't use a light pencil – If the scanner can't see it, I can't grade it. If unsure, ask a proctor.

Please be smart in your time management. This is an assessment.. I want to understand your level of understanding. Don't get stuck on a question. Do the questions you are most sure about first.

Be concise, but be clear and complete. If you make assumptions, write them out. If you need extra space, there are two "scratch" pages at the end of the exam. Please tell me to look there for a continuation of your answer.

Code quality rules still apply (except for comments). So your code must be reasonably performant (e.g. $O(1)$ where it should be $O(1)$), not overly complex, use good names, const, etc...

Show your work, it helps me give you partial credit even if you get lost. I really hate taking off all the points and writing "No attempt made"

You will have until the posted end time to complete the exam.

Desks must be clear. No talking. No notes or electronic devices will be allowed other than a calculator (you shouldn't need one). Don't take pictures of the exam.

If you need to use the restroom, just go – no need to ask. We're adults and we should operate from a level of trust.

For all code, you may assume that we are using `namespace std;` and that we've included all the expected files (`<iostream>`, `<stdexcept>`, `<string>`, `<vector>`, etc...)

If the exam is not on the front table in the box when I call time, then you will receive a zero. Please understand that I truly mean this – I will not tolerate anyone trying to extend the exam. It is not fair to those students who finish the exam on time.

Name: _____

1) For each of the following, write a clear definition

a) Max Heap

b) AVL Tree (please include its most important numeric property)

2a) What are the definition of and the properties of a BST

2b) What are the three primary components that allow us to construct a Hash Table

Name: _____

3) Write `percolate_up`. Note that `index` is the offset where the initial value to percolate up is found

```
class MaxHeap {
    vector<int> array;
    void percolate_up(int index);
public:
    void insert(int key) {
        array.push_back(key);
        percolate_up(array.size()-1);
    }
};
// Percolate the value at i upward in the heap until it hits the root
void MaxHeap::percolate_up(int i) {
```

Name: _____

4) For each of the following, tell me the Big O and why you choose that value. You will need to describe each of the significant parts of the algorithm in sufficient detail to account for the work. Detail what you are counting. Don't be vague (like only saying "height of the tree!")

a) Heapsort

b) Linked list append with only a head

c) Stack Push (array-based with resize)

d) Heap insert

Name: _____

5) Write the code to the prototype for the function I describe below. It should be suitable for insertion into a header file. The implementation will live elsewhere, you are not expected to write it.

- Read words from a named file into a vector and tell me the index of the longest one.

6) Why do we have the rule of three?

7) Why (and how) do we check input files and extracted values (input >> x) for validity?

8) Describe (is pseudo code) an optimal algorithm to find the 3 largest values in a max heap. Please give its Big O and justify it.

Name: _____

9) Complete this class with the 5 standard functions. You are responsible for writing the appropriate const and non-const methods. Do not change the private variables. You are responsible for writing the appropriate const methods and **destructors** (if needed). You do NOT need to conform to the rule of three (other than the destructor). I would like code for a Stack of strings written with a linked list (include **destructor**). I've included the node type:

```
class Stack {
    struct Node* {
        string key;  Node* next;
        Node(const string& key): key(key), next(nullptr); }
    };
    Node* head = nullptr;
public:
```

Name: _____

10) Complete the iterative implementation for `insert` for a **2-3-4** tree. Assume that `split` and `is_leaf` are written for you (and destructors and rule of three, etc...) Just finish `insert`.

```
class BTree {
    struct Node {
        string A,B,C;
        int keycount=0;
        Node *parent=nullptr,*left=nullptr,*mid1=nullptr,*mid2=nullptr,*right=nullptr;
    };
    Node* root = nullptr;
public:
    // These are provided and are assumed to work.
    // split takes a node and hands back a pointer to the new node
    // from which to continue
    Node* split(Node* curr);
    bool is_leaf(Node* curr) const;
    void insert(const string& key) {
        if (root == nullptr) {
            root = new Node;
            root->A = key;
            root->keycount = 1;
            return;
        }
        // Complete the insert function.  split() is already written.
```

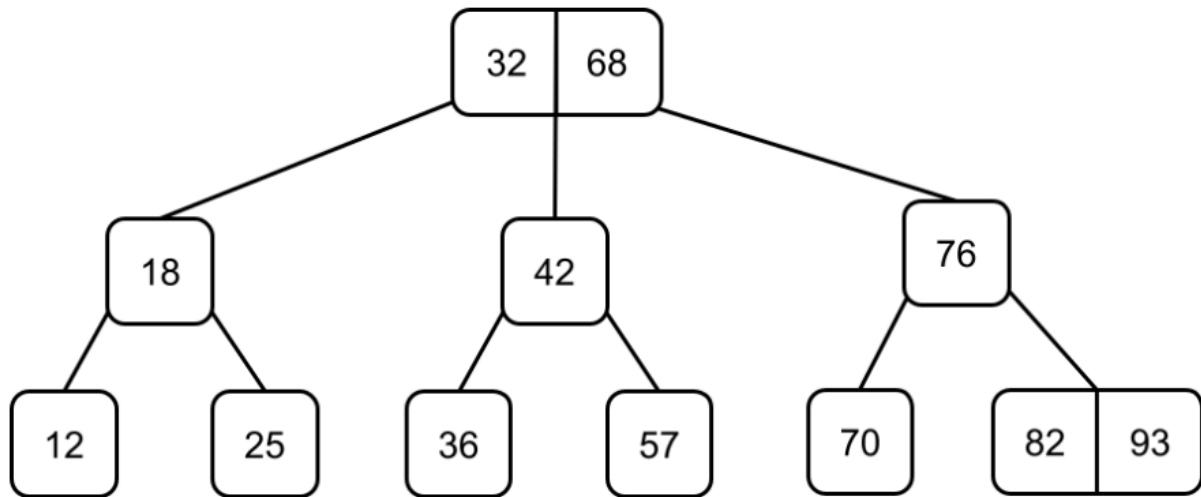
Name: _____

11) Complete the linked list palindrome method started below. A palindrome reads the same forward and backwards. So if the linked list has aa->bb->cc, the palindrome would have aa->bb->cc->cc->bb->aa

```
class LinkedList {
    struct Node {
        string key;
        Node* next;
        Node(const string& key) : key(key), next(nullptr) {}
    };
    Node* head = nullptr;
    Node* tail = nullptr;
public:
    LinkedList() : head(nullptr), tail(nullptr) {}
    LinkedList(const LinkedList& other)
        : head(nullptr), tail(nullptr)
    {
        for(auto* curr=other.head;curr;curr=curr->next) insert(curr->key);
    }
    void insert(const string& key);
    LinkedList palindrome() const { // Makes a copy!
        // The first half of copy is the original linked list
        LinkedList copy = *this;
        if (copy.head != nullptr) {
            // Hint: Create a linked list using reverse_head and
            // reverse_tail then link those to the end of the copy
            Node* reverse_head = nullptr;
            Node* reverse_tail = nullptr;
```

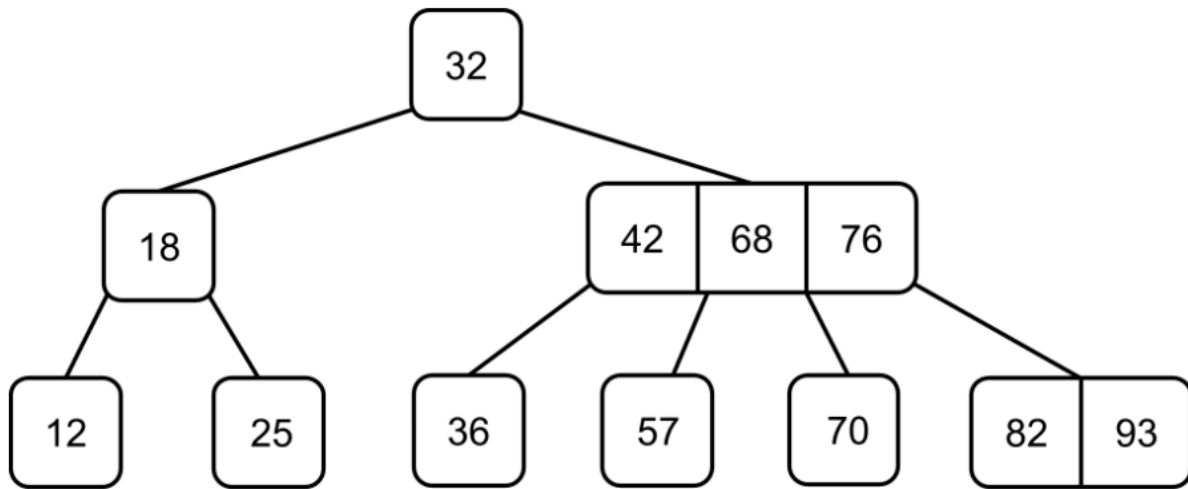

Name: _____

12) Here's a 2-3 tree. Please insert 80 and 81. Draw the resulting tree.



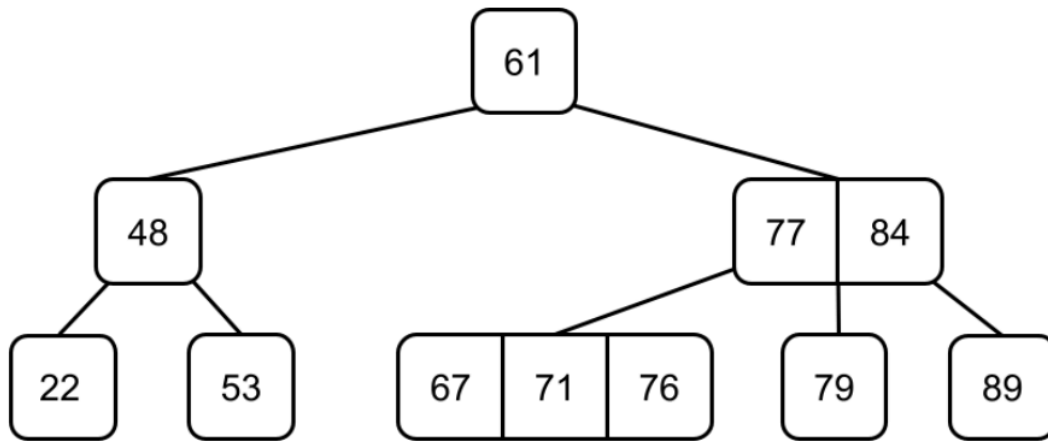
Name: _____

13) Here's a 2-3-4 tree. Please insert 37 and 38. Draw the resulting tree.



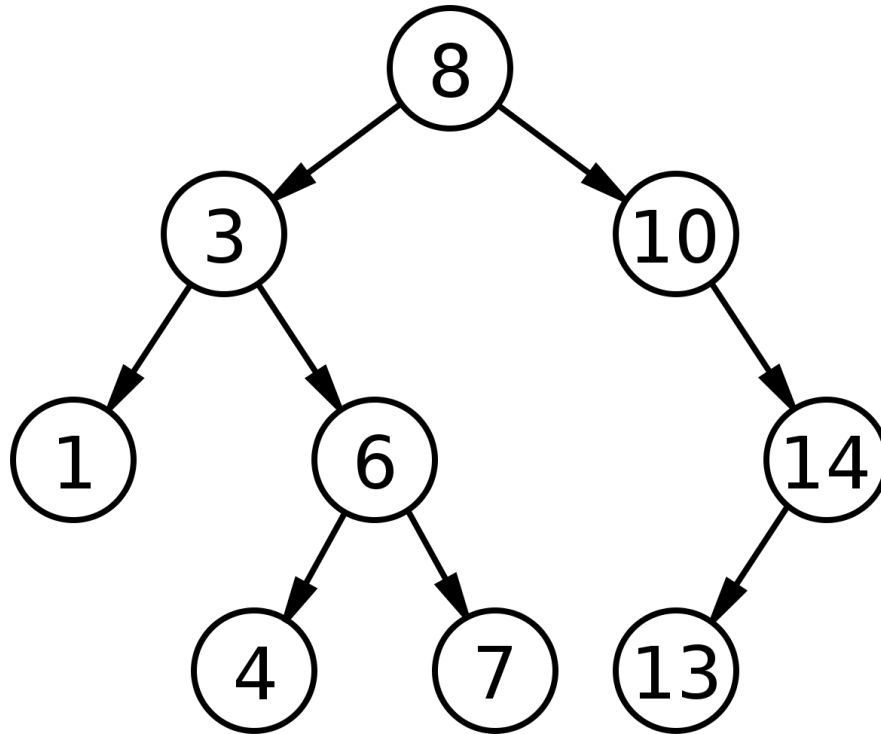
Name: _____

14. Here's a 2-3-4 tree. I want you to draw two trees. First, remove 79 and draw that tree. Then remove 89 and 84. Draw that second tree.



Name: _____

15 Here's an AVL tree. I just added the node with 13. Draw in the balance factor for each node below the key. If not valid, make the appropriate fix and draw below.



Name: _____

16a) For Red-Black trees, what are the rules

16b) Here's a Red-Black tree.

- a) What color is the node labeled [8]
- b) What color is the node labeled [14]
- c) What color is the node labeled [10]
- d) There are 8 possible color combinations for [1] [3] and [6]. If it works, write YES. If a combination doesn't work, say NO and give the reason why.

1:R 3:R 6:R –

1:R 3:R 6:B –

1:R 3:B 6:R –

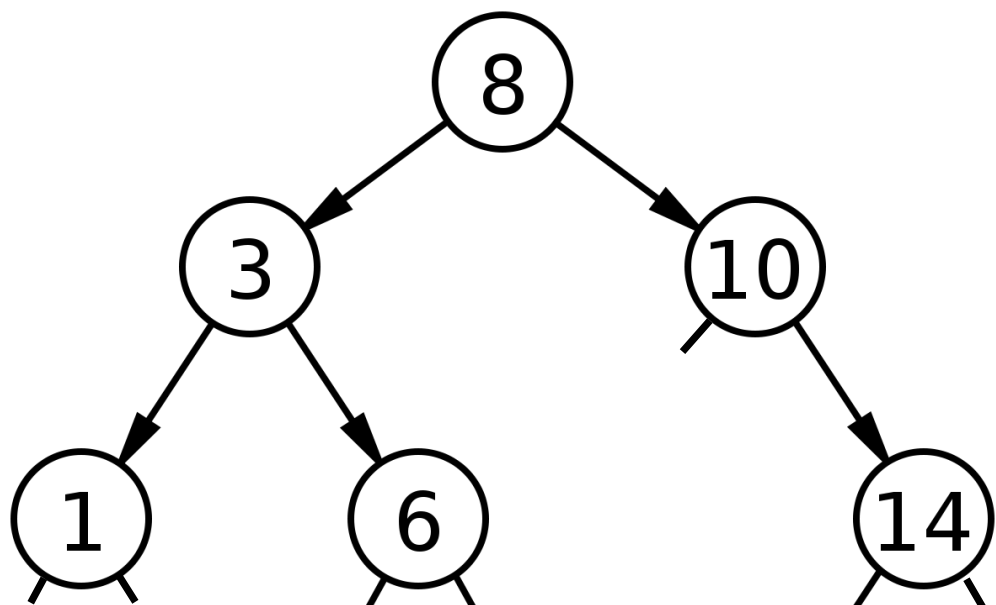
1:R 3:B 6:B –

1:B 3:R 6:R –

1:B 3:R 6:B –

1:B 3:B 6:R –

1:B 3:B 6:B –



Name: _____

17) Linux doesn't store your clear text password, but when you try to log in, the system will let you in when you type the correct one in when prompted. How does that work?

18) I made a mistake creating my hash function and it only returns numbers from 0..50. I use quadratic hashing and have 100 slots in my table. What is going to happen?

19) Insert the following 6 values (in order) into the empty hash table. Your hash function is simply the number modulo 10. Your collision strategy is quadratic probing (both coefficients are 1). It would be a good idea to show your work. **[32, 34, 11, 77, 21, 22]**

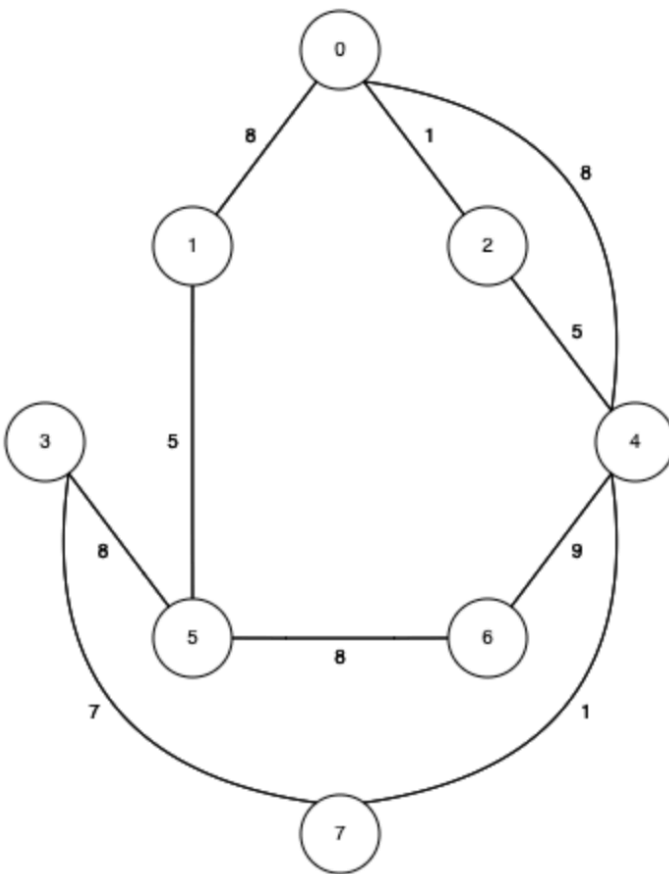
0	
1	
2	
3	
4	
5	
6	
7	
8	
9	

Name: _____

20) What are the steps for Dijkstra's method? Tell me what you do to set it up and what you do at each step.

21) Is Dijkstra a depth first method or a breadth-first approach?

Name: _____



I want you to perform Dijkstra's Method to find all the shortest paths. Note that you are **starting from node 2** (S2 in the table). The V column is simply there as a convenience – it is not graded. The distance and previous node entries ARE graded. You must show your work for credit here. I ask that you cross out values rather than erase or skip them. So, I expect to see things like ~~7~~ 5 3 to indicate that the value was 7, then a distance 5 was found, and then finally 3 was found. Same for the previous nodes.

Label	V	Shortest distance found so far	Previous node in backward path
0			
1			
S2			
3			
4			
5			
6			
7			

Name: _____

Scratch 1

Name: _____

Scratch 2