

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

b) Linked list

2) What is definition of and the properties of a

a) Stack

b) Red-Black tree

Name: _____

3) Write the `mergesort_helper` function. You may assume the `mergesort` and `merge` functions work correctly.

```
void mergesort(vector<int>& array) {
    if (array.size() < 2) return;
    merge_sort_helper(array, 0, array.size()-1);
}
// Merge the low and high partitions back together
void merge(vector<int>& array, unsigned lo, unsigned mid, unsigned hi);
void mergesort_helper(vector<int>& array, unsigned lo, unsigned hi) {
```

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) Mergesort

b) Linked list reverse

c) Queue pop (Linked list)

d) Heap/Priority Queue remove

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 doubles from a named file returning the positive and negative ones in separate vectors passed into the function.

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 (in pseudo code) an optimal algorithm to find the lowest value 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 Queue of strings written with a linked list (include **destructor**). I've included the node type:

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

Name: _____

10) Write `percolate_down`. I will hand you a vector that represents a well formed max heap of size `array.size()` EXCEPT the root value may not **yet** dominate its children. Complete the function so that the array is made to conform to MAX heap rules (done in place)

```
void percolate_down(vector<int>& array) {
```

Name: _____

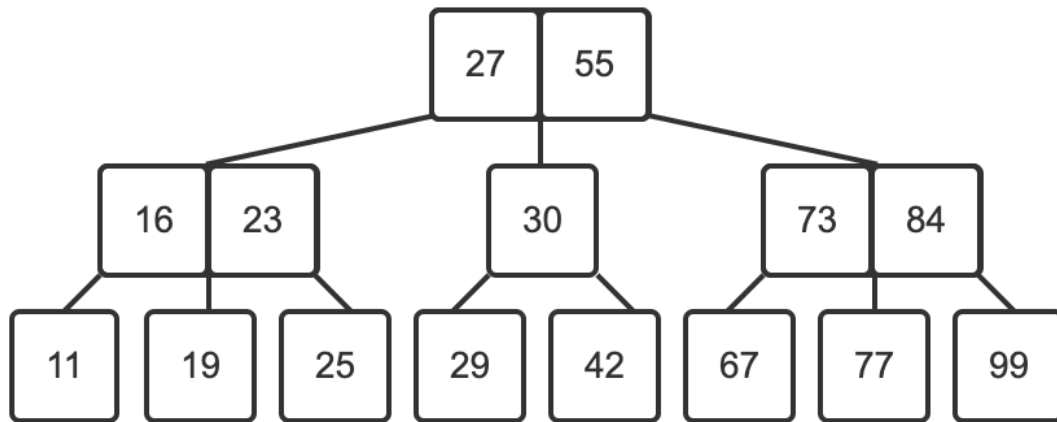
11) Complete the method for set union. Note that it is done IN PLACE (so the values in other are merged with those already in the object). Duplicate keys are ignored since this is a set.

```
class Set {
    struct Node { string key; Node* left; Node* right;
        Node(const string& key) : key(key), left(nullptr), right(nullptr)
    {}
    };
    Node* root;
    // Private helpers for insert and copy constructor not shown
    // You may add things here as needed

public:
    Set();
    ~Set();
    Set(const Set& other); // Copy constructor and rule-of-three done
    Set& operator=(const Set& other);
    void unionWith(Set& other) {
```

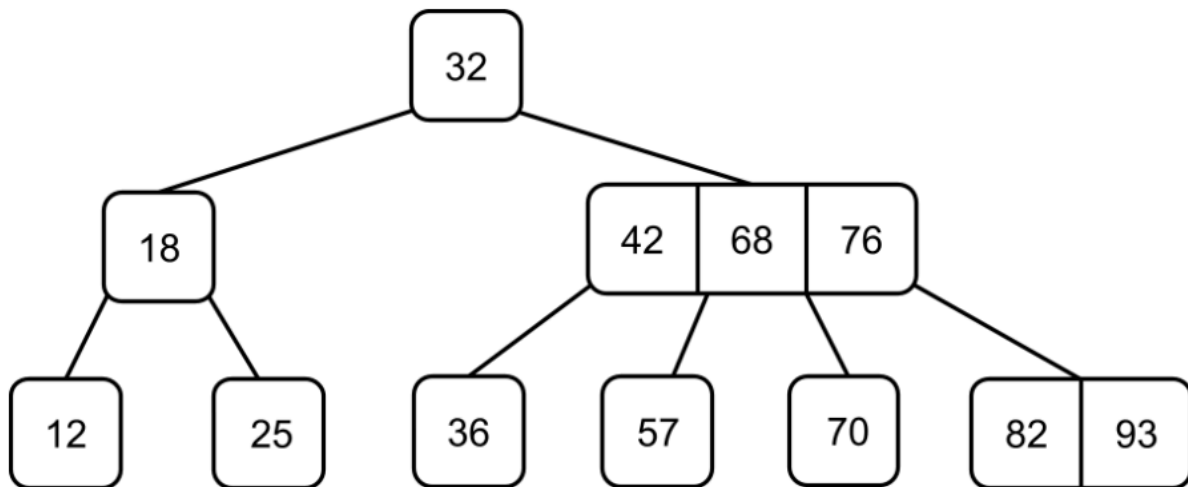

Name: _____

12) Here's a 2-3 tree. Please insert 70 and 71. Draw the resulting tree.



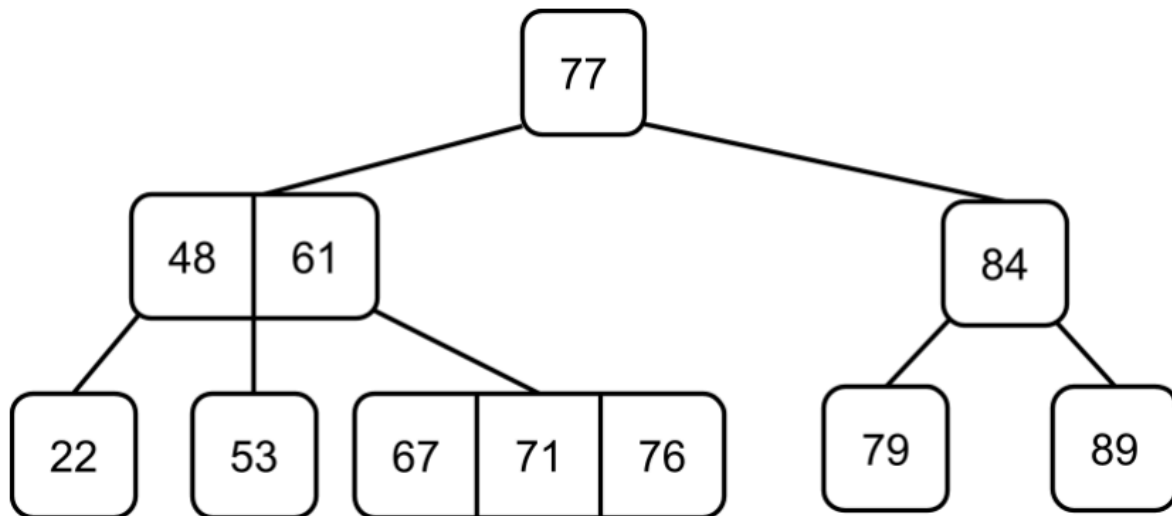
Name: _____

13) Here's a 2-3-4 tree. Please insert 71 and 72. Draw the resulting tree.



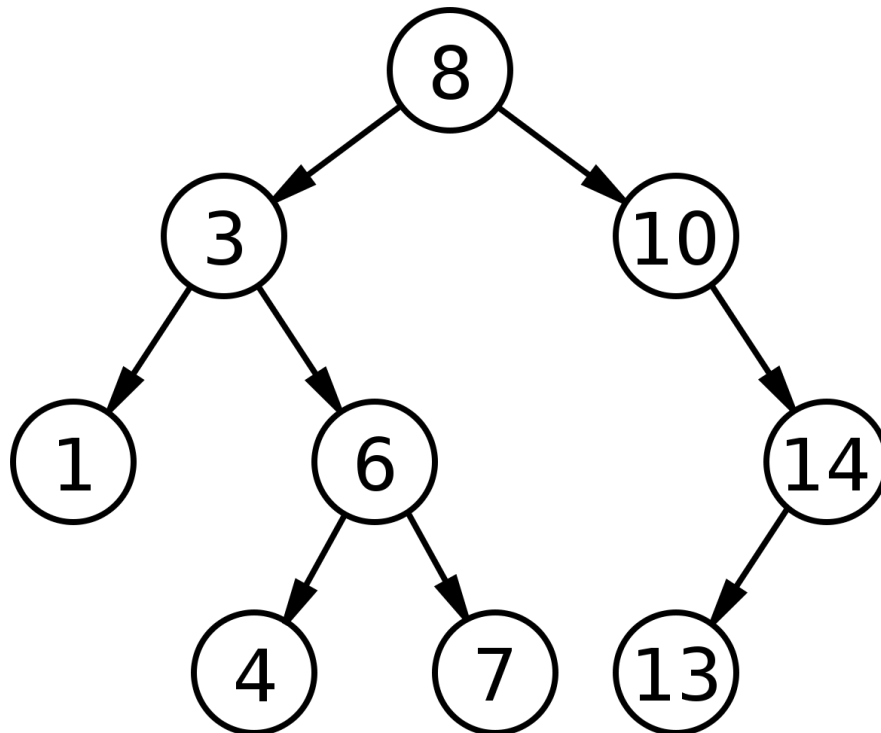
Name: _____

14. Here's a 2-3-4 tree. I want you to draw two trees. First, remove 53 and draw that tree. Then remove 22. 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 –

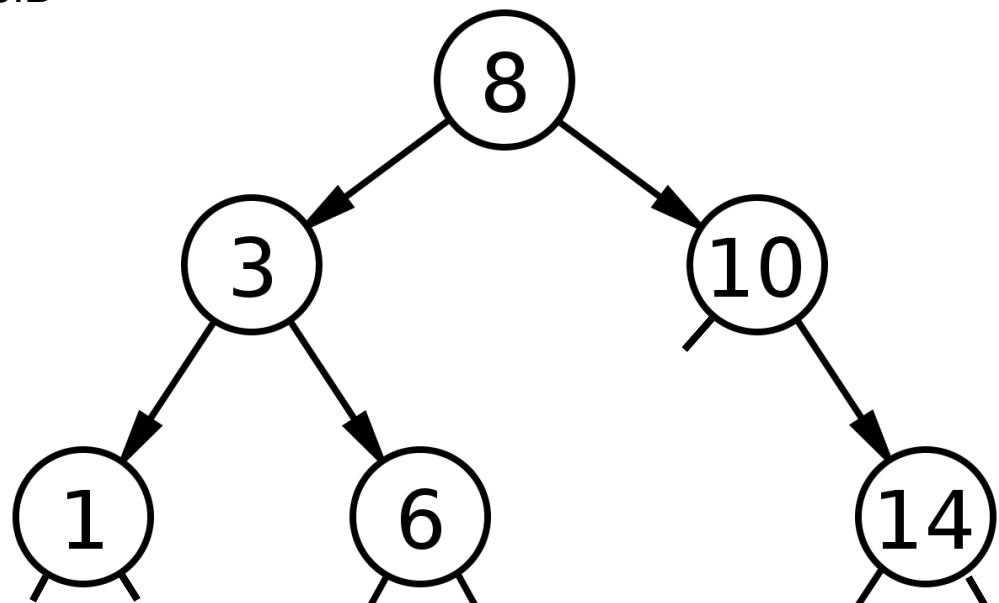
1:R 3:B 6:B –

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 never returns zero. 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. The collision strategy is just quadratic probing (both of the coefficients are 1). It would be a good idea to show your work. **[36, 41, 78, 17, 56, 62]**

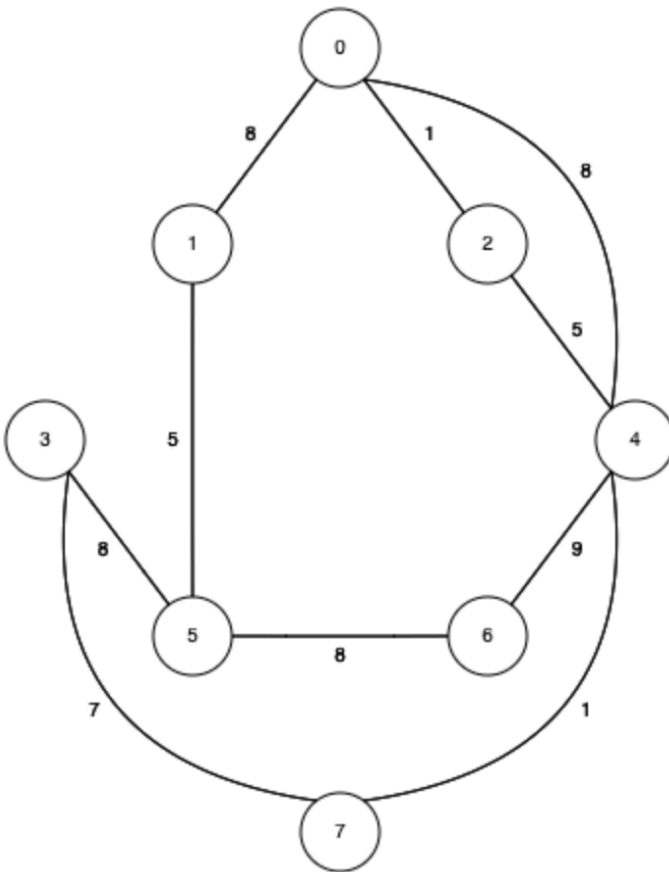
0	
1	
2	
3	
4	
5	
6	
7	
8	
9	

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: _____



22) I want you to perform Dijkstra's Method to find all the shortest paths. Note that you are **starting from node 0** (S0 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
S0			
1			
2			
3			
4			
5			
6			
7			

Name: _____

Scratch 1

Name: _____

Scratch 2