Write your name in box below and NetID to the right ===>	netid
	8am Lecture □
	12:30pm Lecture ☑
Your name goes here	

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 some “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 and rule of three). 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…

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…)

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 (well you can put a stuffed animal or snacks there!). 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, please check out with a proctor.. You must leave your test and phone with them. They will clock you out and back in again.

If you are found talking with other students; I will move you to the front row of the classroom and mark your exam to remind me to check for unauthorized collaboration. No warnings, no second chances.

Similarly, if I see notes or your phone or other direct indications of cheating, I will immediately terminate and confiscate your exam.

If the exam is not on the front table in the box when I call time, then you will receive a zero.

Name: ____

1. [Syllabus] Mark **all** items that are correct regarding my late work policies

[ ] I lose 2% of my grade per day late

[ ] there is ample extra credit available to make up missing work assignments

[ ] Programming and lab assignments are sometimes extended

[ ] Participation and Challenge assignments are routinely extended

2. I have an existing singly linked list built using the node structure below. I want you to write the code that makes a REVERSED COPY of it. So this function will return a head pointer that has the identical keys in its version. So, if the original one was 11 -> 22 -> 33 ->, the new one will be 33 -> 22 -> 11 -> [Hint: Think about where to put 11 and then how to get 22 to point to it]

```
struct Node { int key; Node* next; };
Node* copy_and_reverse_list(const Node* head) {
```

3. Why don't I need to check for a `nullptr` when I do a push_front to a doubly linked list when I have a dummyHead and a dummyTail?

Name: ____

4. Write me a valid prototype for a function (not the implementation, just the part I would put into a header file - DON’T WRITE THE ACTUAL FUNCTION). This function will create a vector of integers and a count and pass both back to me somehow. I’ll pass in the name of a file that I want to read and a vector of integers that I want to exclude from that resulting vector. So, I’ll call it with something like foo.txt and [11,22,33] and get back all the integers in the file that don’t match those numbers and a **count** of the integers that matched. Remember that C++ can only return one value, but I need to retrieve two values after my call.

5. I try to open a file and it succeeds. I immediately check the .eof() flag. What does it tell me?

```
ifstream input(“foo.txt”);
if (not input.is_open()) throw runtime_error(“did not open”);
// we check input.eof() here.  What does it tell me?
```

6. I read a value into an integer and notice it fails. What does this print out?

```
int x = 33;   cin >> x;   if (not cin.good()) cout << x << endl;
```

Name: ____

7. Please finish this class definition for a Stack. Implement the expected functions (e.g. push, pop, top, etc..). I've indicated where to write them on this and the following page. There is overflow room on the last two pages of the exam. Neither change nor add to my private member **variables**. Your implementation should have the right big O.

- If you push when the stack is full, I want it to resize and double the storage rather than fail
- I want one extra function called **reverse** that will create another stack with the values reversed in it. I've started it below
- Please implement the required destructor, I've taken care of the rule-of-three.

```
class Stack {
   int maxsize;
   int size;
   string *array;
   void resize() {



   }
public:
  Stack() : maxsize(10),size(0),array(new string[maxsize]) {}
  Stack(const Stack&) = delete;
  Stack& operator=(const Stack&) = delete;

  Stack reverse() const {
      Stack result; // Start with an empty stack, fill it, return it
```

Code for your reverse in the space above please!

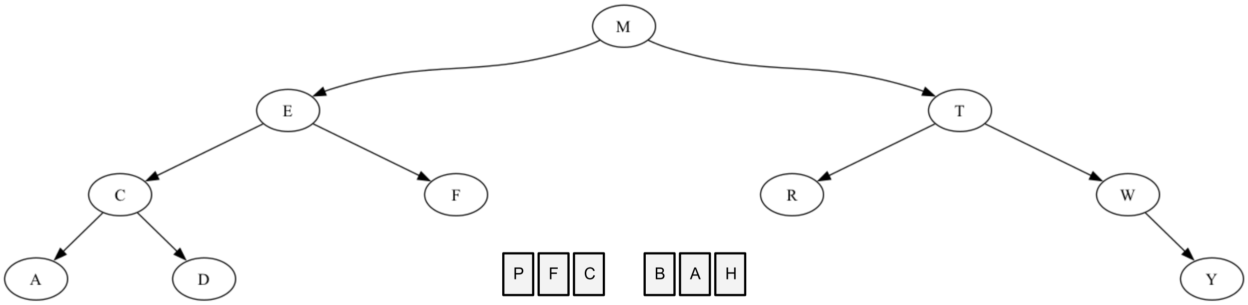
Name: ____

```
// Destructor here




// is_empty here



// push here




// pop here




// all top code here




};
```

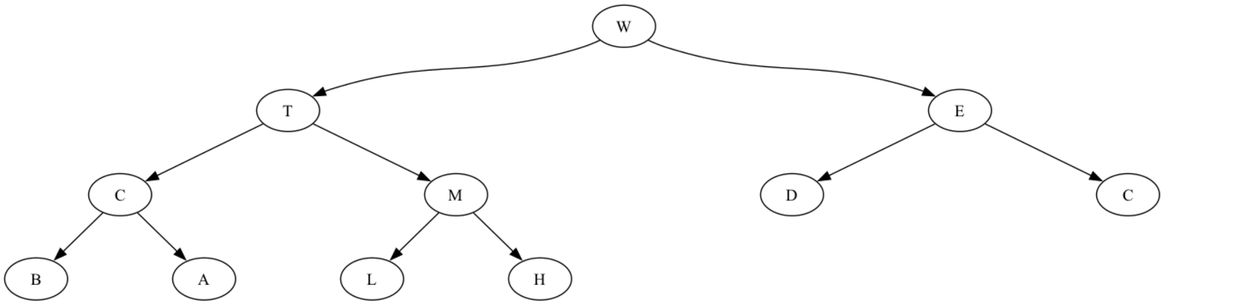
Name: ____

8. Examine the binary trees below. Key order is lexicographical (A < B < C < … < Z)

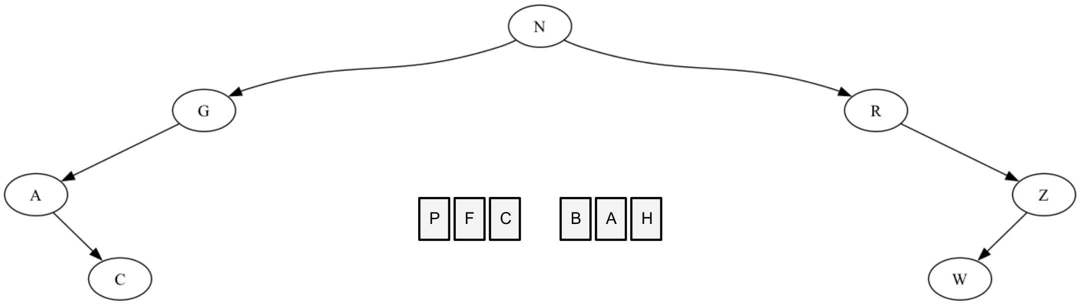
- I want you to classify them by shape as Perfect [P], Full [F], Complete [C] (may be zero or more than one).
- Then classify as BST [B], AVL [A], or MaxHeap [H] (again, select all that apply)



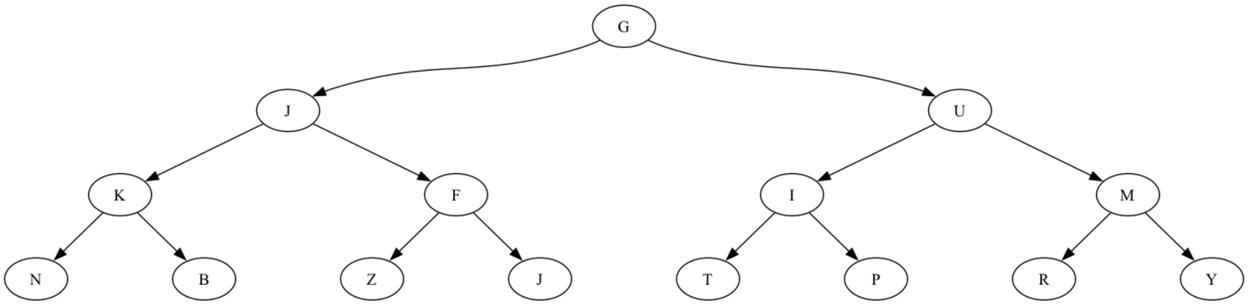
P F C B A H



P F C B A H



P F C B A H



P F C B A H

Name: ____

9. I want you to build and draw a Binary Search Tree (**BST**) by inserting the following values in order: [40, 80, 50, 60, 68, 64, 62]

10. I want you to build and draw a Adelson-Velsky-Landis Tree (**AVL** tree) by inserting those same values in order: [40, 80, 50, 60, 68, 64, 62]

Name: ____

11. I typically write BST::remove recursively so that I can follow up the case where I swap a predecessor/successor node (largest in left subtree, smallest in right subtree) to clean up that new key from my subtree.   The question is… is it possible to write it with loops and not recursion? Please justify your answer.

12. In the Stack you wrote a couple pages back, you implemented Stack::push with resize. Tell me what the big O is and justify your response.

13. So, if I did 1000 stack pushes (in which I clearly did some resizes) and it took 30 milliseconds.I Then I re-ran the program and it did 7000 stack pushes. Approximately how long do you think it would take?

Name: ____

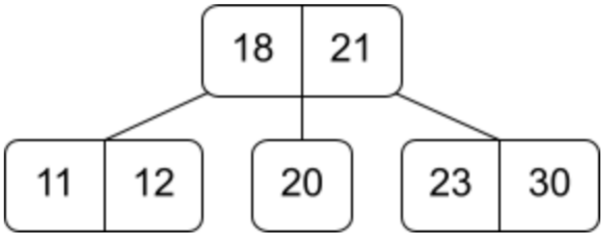
14. Here's the code for a max heap. I'll want you to fill in the code for the unimplemented functions (not the prototypes). It will print out 5 numbers, run your sort, and print out 5 more. Add them to the boxes below in

First 5 □ □ □ □ □ Second 5 □ □ □ □ □

```
#include <iostream>
#include <vector>
using namespace std;
// Don't write these three but you may use them
void percolate_up(int i, vector<int>& array);
void heapinsert(int x, vector<int>& array);
void heapify(vector<int>& array);

void percolate_down(int i, vector<int>& array, int size) {

}
void heapsort(vector<int>& v){ // sort ANY array in place

}
int main() {
  vector<int> v;
  heapinsert(22);heapinsert(15);heapinsert(19);heapinsert(20);heapinsert(26);
  for(const auto& x:v) { cout << x << ' ' << endl; } cout << endl;
  heapsort(v);
  for(const auto& x:5) { cout << x << ' ' << endl; } cout << endl;
  return 0;
}
```

Name: ____

15. Here's a 2-3 tree. I would like you to insert the value 16 and draw that tree

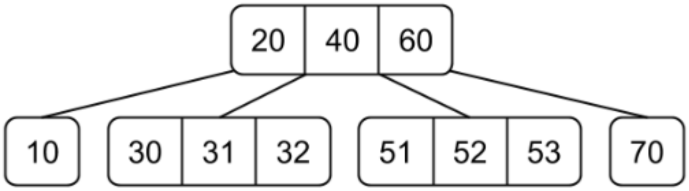


16. Continuing with the answer from question 15, please insert 10.

17. Continue with the answer for question 15, please insert 5

Name: ____

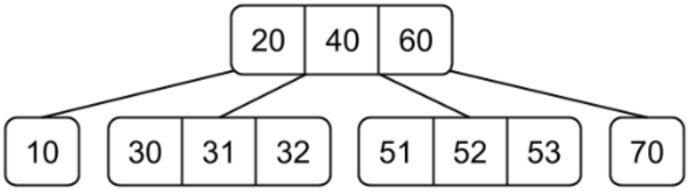
18. Here’s a 2-3-4 tree. I would like you to insert the value 65. Draw the resulting tree.



19. Continue with the result from the question 18 above and insert 55. Draw that tree.

Name: ____

20. Here's that same 2-3-4 tree. Please delete the value 70 and draw the tree.



21. Again, **continue** from the answer from question 20; delete 60 and draw the tree.

22. One more time.. **Continue** from the answer from the question 21, delete 53 and draw **that** tree.

Name: ____

23. What is the definition of (rules for)...

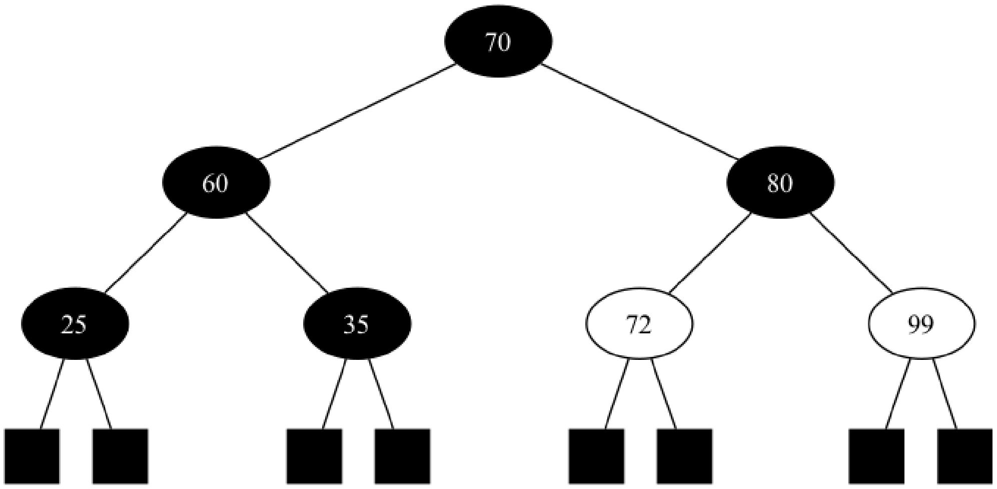
a) An AVL tree

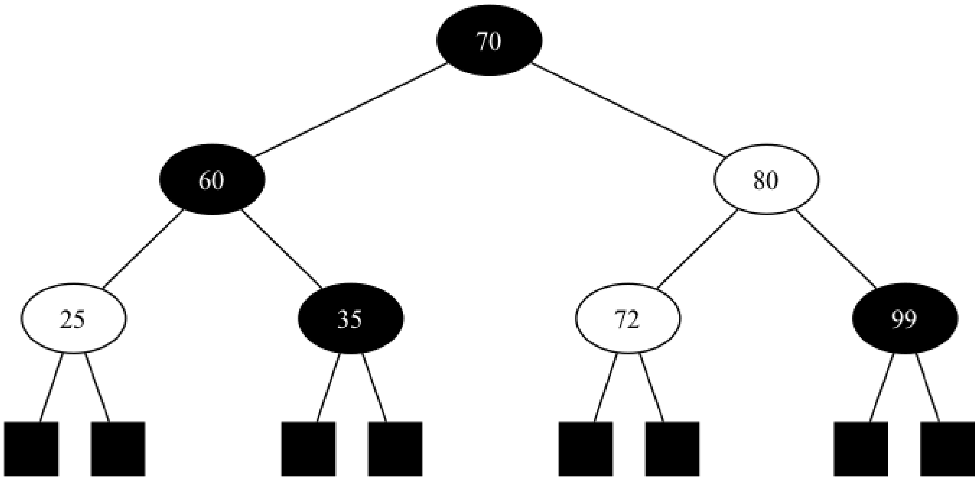
b) A Heap

c) A Red-Black tree

Name: ____

18. For each tree below. (a) Tell me if it is a valid Red-Black tree. (b) If not valid, list **ALL** the reasons it is not a good Red-Black tree (remember your rules from 16(b) on the last page!). **Use the node numbers** in your reasons (e.g. Node 11 is the root and it isn't black).





Name: ____

19. I would like you to explain quicksort to me. Imagine that we are both looking at the source code (you do not need to write it here). I want to know what is going on, I want to know the Big O (in relation to *n*, the size of the array), AND I want to know why you think it is that big O.

20. Your Linux sysadmin cannot recover your password if you lose it because the password is not saved on the system. But if the password isn't there, how does logging in work (assuming you've typed in a correct username and password)?

Name: ____

21. 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!"). Relate the action to the complexity of the work (remember what Big O tells us!) Again, imagine that we are both looking at the code [again, don't write it!] and you are explaining it to me.

a) Mergesort

b) Linked list reverse

c) Queue pop (Linked list)

d) Heap/Priority Queue remove

Name: ____

22. I tell you the three things you need to make a hash table

- The hash function is modulo 10 (so the last decimal digit of a number)
- The slots are either blank (no value) or a number [we aren’t deleting]
- My collision strategy is quadratic probing to find an empty slot (with the standard c_1 and c_2 both set to 1)

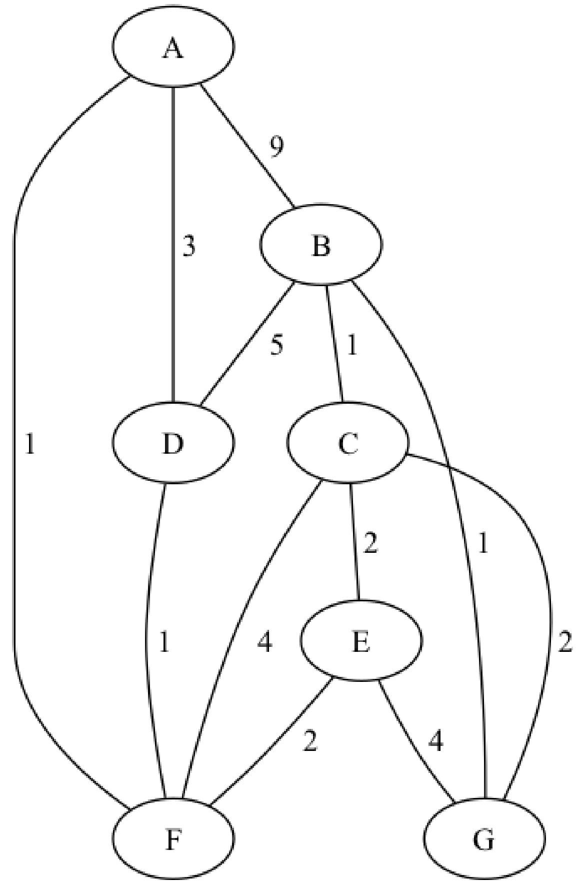
Please insert the values [35, 74, 14, 15, 54, 58, 77, 30, 27] into the hash table so they land in the appropriate slot.

0	
1	
2	
3	
4	
5	
6	
7	
8	
9	

Name: ____

23. Just write out the steps you will take to implement Dijkstra's Method. Once you have the steps, you'll use them for the graph below. Keep careful track of your work since I want to see each step (so cross out old work). Each node needs a final distance from the start node "**A**", and all of the "prev" nodes you found in the backwards path.

24. Fill in this table using the Dijkstra steps (start node A), Please, mark each node visited as you visit it. I need to see **<u>all</u>** your work, so as you update things (e.g. the shortest distance), CROSS OUT THE OLD ONE and then write in the new one.



La bel	V	Shortest distance found so far	Previous node in backward path
A			
B			
C			
D			
E			
F			
G			

Name: ____

Scratch 1

Name: ____

Scratch 2