Create Rubric
85 points

Create your rubric now or come back to it later. You can also make edits to your rubric while grad

Q1
0 points

You have 3 hours to complete the exam from the time you start. DSP students will receive extended time depending on their accommodation.
Save your answers periodically and only submit when you have completed the exam.

This exam is open book, open notes. This includes the following resources:

- zybook or other textbooks
- lecture notes/slides by instructor or your own lecture notes
- code that you wrote for lab/programming assignments
- all code that has been provided on Gauchospace
- https://www.cplusplus.com/reference/

If you need to ask a clarifying question on the exam, make a private post on Piazza.

Do NOT:

- Post any part of the exam on a public forum (including Piazza)
- Ask for help on the exam from an online forum or anyone else.

By selecting the option "yes" below, you are asserting that all work on this exam is yours alone, and that you will not provide any information to

Removing the **Correct** a with auto-grading for thi

1 +0.0
Correct

2 +0.0
Incorrect

+ Add Rubric Item

of this exam with anyone constitutes a violation of the academic integrity agreement for CMPSC24.

I understand the instructions and academic honesty policy for this exam

Yes

No

Q2 Heaps

8 points

Consider the following vector representation of a priority queue configured as a min-Heap with keys 10, 88, 14, 32 , and 51 (pushed to the heap in that order).

Answer the questions that follow.

Q2.1

2 points

What are the key values of the children of key 32? If key 32 does not have a left or right child, put a dash(-)

Left child:

88

Right child:

51

⚠ Removing the **Correct** a with auto-grading for thi

1 +2.0

Correct - 88, 55

2 +0.0

Incorrect

3 +2.0

✚ Add Rubric Item

Q2.2

4 points

⚠ Removing the **Correct** a with auto-grading for thi

Write the elements of the vector representing the min-Heap at each of the following indices after popping the minimum value.

1 +4.0
Correct

Index 0:

14

2 +0.0
Incorrect

Index 1:

32

✚ Add Rubric Item

Index 2:

51

Index 3:

88

Q2.3

2 points

⚠ Removing the **Correct** a with auto-grading for thi

What are the left and right children of root after performing the pop operation on the original min-

Left child:

32

Right child:

51

1	+2.0 Correct - 32, 51
2	+0.0 Incorrect
3	+2.0 Correct

✚ Add Rubric Item

Q3

19 points

Below you are given a partially complete definition of a class Car:

```
#include <vector>
#include <iostream>
#include <set>
#include <string>

using namespace std;
class Car{
   private:
      string manufacturer;
      string model;
      double cost;  // cost in dollars
      double maxspeed; //miles per hour
   public:

       Car(string manufacturerName, string modelName, double costDollars,
         double maximumSpeed): _____ /*A*/, _____/*B*/,
         _____/*C*/, _____/*D*/ _____/*E*/    //Question 1
      ~Car(){}
      string getManufacturer() const{return manufacturer;}
      string getModel() const {return model;}
```

```
        double getmaxSpeed() const{return maxspeed;}
};


class compareSpeeds{
public:
   bool __________/*A*/(__________/*B*/ first, __________/*C*/ second ){  // Question 2
      return first->getmaxSpeed() > second->getmaxSpeed();
   }
};

void sortCars(vector<Car*>& v, compareSpeeds){
      set<Car*, compareSpeeds> s; //line 1
      for(auto elem: v){   //line 2
          s.insert(elem);   //line 3
      }
      int i = 0;            //line 4
      for( auto elem: s){    //line 5
          v[i] = elem;      //line 6
          i++;              //line 7
      }
}

int main(){

  /**  Code to create a vector of cars **/

   __________ compare;
 Question 3
   sortCars(cars, compare);

   return 0;
}
```

Q3.1

1 point

Question 1 - Fill in the blanks to create a parameterized constructor for Car (omit any spaces)

⚠ Removing the **Correct** a with auto-grading for thi

Question 1 - A:

1 +1.0

manufacturer(manufacturerName)

2 +1.0
Correct

3 +0.0
Incorrect

✚ Add Rubric Item

Q3.2
1 point

Question 1 - B:

⚠ Removing the **Correct** a
with auto-grading for thi

model(modelName)

1 +1.0
Correct

2 +0.0
Incorrect

✚ Add Rubric Item

Q3.3
1 point

Question 1 - C:

⚠ Removing the **Correct** a
with auto-grading for thi

1 +1.0

cost(costDollars)

2 +0.0
Incorrect

✚ Add Rubric Item

Q3.4
1 point

Question 1 - D:

maxspeed(maximumSpeed)

⚠ Removing the **Correct** a
with auto-grading for thi

1 +1.0
Correct

2 +0.0
Incorrect

✚ Add Rubric Item

Q3.5
1 point

Question 1 - E:

{}

⚠ Removing the **Correct** a
with auto-grading for thi

1 +1.0
Correct

2 +0.0

✚ Add Rubric Item

Q3.6

1 point

⚠ Removing the **Correct** a with auto-grading for thi

Question 2 - Fill in the blanks to create a comparison class for comparing top speeds

Question 2 - A:

operator()

1 +1.0
Correct

2 +0.0
Incorrect

✚ Add Rubric Item

Q3.7

1 point

⚠ Removing the **Correct** a with auto-grading for thi

Question 2 - B:

Car*

1 +1.0
Correct

2 +0.0
Incorrect

✚ Add Rubric Item

Q3.8

1 point

Question 2 - C:

Car*

⚠ Removing the **Correct** a
with auto-grading for thi

1 +1.0
Correct

2 +0.0
Incorrect

✚ Add Rubric Item

Q3.9

1 point

Question 3 -

compareSpeeds

⚠ Removing the **Correct** a
with auto-grading for thi

1 +1.0
Correct

2 +0.0
Incorrect

✚ Add Rubric Item

Q3.10

2 points

Which line in the sortCars() function calls the copy constructor of the class Car. Select all that apply.

- line 1
- line 2
- line 3
- line 4
- ✓ line 5
- line 6
- line 7
- none of the above (copy constructor is never called)

⚠ Removing the **Correct** a
with auto-grading for thi

1	+2.0 Correct: line 6
2	+2.0 Correct
3	+1.0 line 6 and one other ir
4	+0.0 Incorrect

✚ Add Rubric Item

Which line in the sortCars() function calls the copy assignment of the class Car. Select all that apply.

line 1

line 2

line 3

line 4

line 5

✓ line 6

line 7

none of the above (copy assignment is never called)

Q3.11

5 points

What does the above code do?

⚠ Removing the **Correct** a
with auto-grading for thi

1 +2.0

Correct: Car type obje
which is needed by th
operator for the Car cl

2 +5.0

Correct

3 +0.0

Incorrect

4 +1.0

Creates a vector of cars that gets sorted from lowest to highest speed

Creates a vector of car pointers that gets sorted from lowest to highest speed

Creates a vector of cars that gets sorted from highest to lowest speed

✓ Creates a vector of car pointers that gets sorted from highest to lowest to speed

None of the above

specific about which opera

5 +0.5

Incorrect reason: copy generate a default vers the reason for the com

6 +0.0

Incorrect reason: class reasoning related to te

7 +0.0

Incorrect reason: line default constructor of of fooCars() calls the Car OR simply stated

8 +0.0

Incorrect: Some other destructor OR shallow

9 +0.0

Incorrect reason: fooC or need a member fun const.

0 +0.0

Missing or completely

✚ Add Rubric Item

Q3.12

3 points

⚠ Removing the **Correct** a with auto-grading for thi

What is the big-O of sortCars() in terms of N cars?
Follow the following guidelines, YOU WILL LOSE

ANSWER

Write exponents as "x^y"
Log base 2 as "log(x)"
Multiplication as "xy" (No asterisk*)
OMIT ALL SPACES
NOTE - N is capitalized (Don't forget the "O"!)
Example: "O(MN^2)" or "O(log(N))"

O(Nlog(N))

1	+2.0 Correct: Looking for a or non member functi bool operator<(const return lhs.getCost() } If non member func it must be declared as
2	+3.0 Correct
3	+0.0 Incorrect
4	+1.5 Non-member function
5	+1.5 Correct implementatic
6	+0.0 Incorrect implementat
7	+0.0 Missing or completely

✚ Add Rubric Item

Q4
19 points

Answer the following questions given this code:

```
class smartQueue{
  private:
    queue<int> mainQ;
    ___________ maxQ;  // Question 1-A
    ___________ minQ;  // Question 1-B

  public:
    void push(int value);
    void pop();
    bool empty();
    int min();
    int max();
    int front();
};

void smartQueue::push(int value)
{
  mainQ.push(value);
  if(___________)              maxQ.push_back(value);    // Question 1-C
  else
    if(maxQ.back()>=value)          maxQ.push_back(value);
    else{
      while(!maxQ.empty() && maxQ.back()<value) maxQ.pop_back();
      maxQ.push_back(value);
    }

  if(___________)              minQ.push_back(value);    // Question 1-D
  else
    if(minQ.back()<=value)          minQ.push_back(value);
    else{
      while(!minQ.empty() && minQ.back()>value) minQ.pop_back();
      minQ.push_back(value);
    }
}

void smartQueue::pop()
{
  int value = mainQ.front();
  mainQ.pop();
  if(value == maxQ.front()) maxQ.pop_front();
  if(value == minQ.front()) minQ.pop_front();
}

int smartQueue::max()
{
  return ___________;       Question 1-E
```

```
int smartQueue::min()
{
   return _____;        Question 1-F
}

int smartQueue::front()
{
   return _____;        Question 1-G
}

bool smartQueue::empty()
{
   return _____;        Question 1-H
}
```

Q4.1

1 point

Question 1 - Fill in the blanks in the above code

A:

deque<int>

⚠ Removing the **Correct** a
with auto-grading for thi

1 -0.0
Correct: For reference
s21/final-exam-solutic

2 -0.0
Correct

3 -1.0
Incorrect

4 -5.0
Implemented a smartS

5 -3.0

front(). For example. after
be 1.

6 -3.0

pop() incorrect: first e
popped

7 -3.0

min or max incorrect:
This implementation
current min element i
min() - should return
single queue that stor
issue is that the min/m
on the values in the qu
the queue but also on
is popped. So, the pro
the smartStack proble

8 -3.0

push incorrect

9 -1.0

empty() incorrect or n

0 -2.0

Runing time of two or

-2.0

min/max partially cor
keep track of the min/
min/max values are st
pushing 4, 1, 7, 1 is t
expected). But when
is removed from the f
removed from the pric
elements. If we called
more time, followed b
4 as a result of this log
involves used stacks t

-1.0

-1.0
See comments (compi

-2.0
See comments (logic

-5.0
On the right track but

-10.0
blank/incorrect

✚ Add Rubric Item

Q4.2
1 point

B:

deque<int>

⚠ Removing the **Correct** a
with auto-grading for thi

1 +1.0
Correct

2 +0.0
Incorrect

✚ Add Rubric Item

Q4.3
1 point

C:

maxQ.empty()

Removing the **Correct** a
with auto-grading for thi

1 +1.0
Correct

2 +0.0
Incorrect

+ Add Rubric Item

Q4.4
1 point

D:

minQ.empty()

Removing the **Correct** a
with auto-grading for thi

1 +1.0
Correct

2 +0.0
Incorrect

+ Add Rubric Item

Q4.5
1 point

E:

maxQ.front()

Removing the **Correct** a
with auto-grading for thi

1 +1.0
Correct

2 +0.0
Incorrect

✚ Add Rubric Item

Q4.6
1 point

F:

minQ.front()

⚠ Removing the **Correct** a
with auto-grading for thi

1 +1.0
Correct

2 +0.0
Incorrect

✚ Add Rubric Item

Q4.7
1 point

G:

mainQ.front()

⚠ Removing the **Correct** a
with auto-grading for thi

1 +1.0
Correct

2 +0.0
Incorrect

✚ Add Rubric Item

Q4.8
1 point

⚠ Removing the **Correct** a with auto-grading for thi

H:

mainQ.empty()

1 +1.0
Correct

2 +0.0
Incorrect

✚ Add Rubric Item

Q4.9
1 point

⚠ Removing the **Correct** a with auto-grading for thi

What is the big-O of the following functions based off N entries in smartQueue? Follow the following guidelines, YOU WILL LOSE 50% FOR INCORRECTLY FORMATTING YOUR ANSWER

Write exponents as "x^y"
Log base 2 as "log(x)"
Multiplication as "xy" (No asterisk*)
OMIT ALL SPACES

1 +1.0
Correct

2 +0.0
Incorrect

Example: "O(MN^2)" or "O(log(N))"

push:

O(N)

Q4.10

1 point

pop:

O(1)

Removing the **Correct** a with auto-grading for thi

1 +1.0
Correct

2 +0.0
Incorrect

+ Add Rubric Item

Q4.11

1 point

max:

O(1)

Removing the **Correct** a with auto-grading for thi

1 +1.0
Correct

2 +0.0
Incorrect

✚ Add Rubric Item

Q4.12

1 point

⚠ Removing the **Correct** a with auto-grading for thi

min:

O(1)

1 +1.0
Correct

2 +0.0
Incorrect

✚ Add Rubric Item

Q4.13

1 point

⚠ Removing the **Correct** a with auto-grading for thi

front:

O(1)

1 +1.0
Correct

2 +0.0
Incorrect

✚ Add Rubric Item

Q4.14

empty:

O(1)

⚠ Removing the **Correct** a with auto-grading for thi

1 +1.0
Correct

2 +0.0
Incorrect

✚ Add Rubric Item

Q4.15

5 points

What is the average time complexity of the push operation assuming randomly pushed numbers?

⚠ Removing the **Correct** a with auto-grading for thi

1 +5.0
Correct

2 +0.0
Incorrect

✚ Add Rubric Item

About the same as the worst case Big-O for push

About one half of the worst case Big-O for push

✓ Significantly less than the Big-O for push

About N/2

O(1)

More than one of the above

None of the above

Q5

8 points

Find the tightest worst case Big-O running time of each of the following code as a function of the input size N.

Write exponents as "x^y"
Log base 2 as "log(x)"
Multiplication as "xy" (No asterisk*)
OMIT ALL SPACES
NOTE - N is capitalized (Don't forget the "O"!)
Example: "O(MN^2)" or "O(log(N))"

Q5.1

2 points

```
for(var i=1; i<M; i++)
  i*=N
```

O(M/N)

✓ $O(log_N(M))$

$O(log_M(N))$

O(N)

O(M)

⚠ Removing the **Correct** a
with auto-grading for thi

1	-0.0 Correct: O(n)
2	-0.0 Correct
3	-2.0 Incorrect
4	-3.0 incorrect big O worst
5	-1.5 no/incorrect/not cpmp

✚ Add Rubric Item

Q5.2

3 points

```
int a, b, c = 0;
for (a = N / 2; a <= N; a++) {
  for (b = 2; b <= N; b = b * 2) {
    c = c + N / 2;
  }
}
```

Big O:

⚠ Removing the **Correct** a
with auto-grading for thi

1	-0.0 correct: O((n^2)log(n) the students: The first logN times, because tl linearly. Within each at an upper bound of sequentially to N each would push to the stac since the push functio Now, the while loop n logN items in the stac

O(Nlog(N))

decreases linearly. So, this
upper bounded by N*N log
constant time.
The total upper-bounded ru
logN + N^2 logN + 1 (cout

2 -0.0
Correct

3 -3.0
Incorrect

4 -1.5
correct explanation fo
for second

5 -2.0
incorrect explanations

6 -3.0
incorrect/blank

✚ Add Rubric Item

Q5.3

3 points

⚠ Removing the **Correct** a
with auto-grading for thi

```
int foo(vector<int>& v, int N){ //Assume v has N elements,
                                // where N> 100
  priority_queue<int> p;
  for(auto i:v){
    p.push(i);
  }
  int j = 0;
  while(!p.empty() && j < 100){
```

1 -0.0
Correct

2 -3.0

```
        j++;
      }
      return p.top();
    }
```

Big O:

O(Nlog(N))

3	-3.0 Wrong about the com
4	-4.0 Wrong justification or
5	-2.0 Wrong Big-O. Wrong
6	-2.0 Wrong justification or
7	-2.0 Correct Big-O, but ins

✚ Add Rubric Item

Q6

10 points

Insert the following keys (in the order provided from left to right) into a BST that is initially empty:

4, 7, 20, 2, 50, 9, 10

Answer the questions that follow about the resulting BST

Q6.1

What is the sequence of keys printed in a pre-order traversal of the BST?
Separate the keys by commas, no spaces!

4,2,7,20,9,10,50

⚠ Removing the **Correct** a with auto-grading for thi

1 +0.0
Correct: 4,2,7,20,9,10

2 +2.0
Correct

3 +0.0
Incorrect

4 +1.0
salt

✚ Add Rubric Item

Q6.2
2 points

What is the height of the resulting tree?

4

⚠ Removing the **Correct** a with auto-grading for thi

1 +1.0
Correct: 4

2 +0.0
Incorrect

3 +2.0

+ Add Rubric Item

Q6.3

2 points

Which of the following are leaf nodes. SELECT all that apply

- ✓ 2
- 4
- 7
- 9
- ✓ 10
- 20
- ✓ 50

⚠ Removing the **Correct** a with auto-grading for thi

1 +2.0
Correct

2 +0.0
Incorrect

+ Add Rubric Item

Q6.4

2 points

Where is the predecessor of node with key '7'?

⚠ Removing the **Correct** a with auto-grading for thi

1 +2.0
Correct

In 7's left subtree

7's parent

7's grandparent (parent's parent)

Does not exist

2 +0.0
Incorrect

✚ Add Rubric Item

Where is the successor of node with key '7'

In 7's right subtree

In 7's left subtree

7's parent

7's grandparent (parent's parent)

Does not exist

Q6.5

2 points

⚠ Removing the **Correct** a with auto-grading for thi

Where is the predecessor of node with key '20'?

In 20's right subtree

In 20's left subtree

20's parent

20's grandparent (parent's parent)

Does not exist

1 +2.0
Correct

2 +0.0
Incorrect

Where is the successor of node with key '20'

✚ Add Rubric Item

In 20's left subtree

20's parent

20's grandparent (parent's parent)

Does not exist

Q7

6 points

Now insert the same keys in the same order into an AVL Tree (a type of self balancing BST)

4, 7, 20, 2, 50, 9, 10

Answer the following questions

Q7.1

2 points

What is the height of the tree?

3

⚠ Removing the **Correct** a with auto-grading for thi

1 +2.0
Correct

2 +0.0
Incorrect

✚ Add Rubric Item

Q7.2

2 points

Where is the in-order predecessor of node with key 7?

- In 7's right subtree
- In 7's left subtree
- 7's parent
- 7's grandparent (parent's parent)
- Does not exist

Where is the in-order successor of node with key '7'

- In 7's right subtree
- In 7's left subtree
- 7's parent
- 7's grandparent (parent's parent)
- Does not exist

⚠ Removing the **Correct** a with auto-grading for thi

1 +2.0
Correct

2 +0.0
Incorrect

✚ Add Rubric Item

Q7.3

2 points

Where is the in-order predecessor of node with key 20?

- In 20's right subtree
- In 20's left subtree
- 20's parent
- 20's grandparent (parent's parent)
- Does not exist

Where is the in-order successor of node with key '20'

⚠ Removing the **Correct** a with auto-grading for thi

1 +2.0
Correct

2 +0.0
Incorrect

In 20's left subtree

20's parent

20's grandparent (parent's parent)

Does not exist

Q8

5 points

Given the function func, choose the output of the function when given an input of 53.

Note that the function prints the output within its body and assume all required header files are included

```
void func(int n)
{
   stack<int> S;
   while (n > 0)
   {
    S.push(!(n%2));
    n = n/2;
   }
   while (!S.empty())
   {
      cout << S.top();
      S.pop();
   }
}
```

001010

Removing the **Correct** a with auto-grading for thi

1 +5.0
Correct

2 +0.0
Incorrect

+ Add Rubric Item

5 points

Assuming the following definition for a node of a binary tree.

```
struct Node{
   int num;
   Node* left;
   Node* right;
};
```

Complete the following function so that executing findMaxInRow(root) returns the maximum number of Nodes at any particular height of the BST. Assume all libraries and constructors are properly initialized.

Note that semicolons are NOT a part of your response.Do NOT put SPACES in between words.

```
int findMaxInRow(Node* root) {
   int maxWidth = 0;
   queue<Node *> q;

   if (root != NULL)
      q.push(root);
   while (true) {
      int queue_len = q.size();

      if (queue_len > maxWidth)
         maxWidth = queue_len;

      for (int i = 1; i <= queue_len; i++) {
         Node *ptr = _____ ;  // Blank #1
         q.pop();

         if (ptr->left != NULL)
            q.push(ptr->left);
         if (ptr->right != NULL)
            q.push(ptr->right);
```

```
        if (_____) // Blank #2
            return maxWidth;

    }
}
```

Q9.1

2 points

Provide your response for blank #1

q.front()

Removing the **Correct** a with auto-grading for thi

1 +2.0
Correct

2 +0.0
Incorrect

+ Add Rubric Item

Q9.2

2 points

Provide your response for blank #2

q.empty()

Removing the **Correct** a with auto-grading for thi

1 +2.0
Correct

2 +0.0
Incorrect

+ Add Rubric Item

Q9.3

1 point

What is wrong with the code above (assuming the blanks are properly filled in)?

- It produces a memory leak
- It can result in a segmentation fault
- It will definitely result in a segmentation fault
- There is nothing wrong with the above code

⚠ Removing the **Correct** a with auto-grading for thi

1 +1.0
Correct

2 +0.0
Incorrect

✚ Add Rubric Item

Q10

5 points

```
int climbStairs(int totalStairs){
    // create a Memoization array for
    // storing the recursion results at each step
    int memo[totalStairs+1];
    //initialize all memo elements to 0
    for(int i = 0; i < totalStairs+1; i++){
        memo[i] = 0;
    }
    return recursionHelper(0, totalStairs, memo);
}

// Pass in the current stairs and total number of stairs.
int recursionHelper(int i, int total, int *memo) {
    if (i > total) {
        return 0;
    }
```

⚠ Removing the **Correct** a with auto-grading for thi

1 +5.0
Correct

2 +0.0
Incorrect

✚ Add Rubric Item

```
        return 0;
    }
    if (memo[i] > 0) {
        return i;
    }
    memo[i] = recursionHelper(i+1, total, memo) + \
        recursionHelper(i+2, total, memo);
    return memo[i];
}
```

In the above code from the practice final, what is the purpose of the variable "memo"?

- It is used to reduce the time complexity
- It is used to reduce the space complexity
- It doesn't reduce space or time complexity
- It reduces both space and time complexity