3013 - Algorithms
Spring 2020
Final Exam
May 12, 2020

Name: _____

READ THESE INSTRUCTIONS
• Create a digital document that has zero handwriting on it. • Your presentation and thoroughness of answers is part of your grade. I make every effort to create clear and understandable questions. You should do the same with your answers. • Questions should be answered in order and clearly marked. • Your name should be on each page, in the heading if possible. • Turn your final in by uploading to D2L. Link should be obvious on D2L course main page.
Failure to comply will result in loss of letter grade

Grade Table (don't write on it)

Question	Points	Score
1	20	
2	50	
3	20	
4	20	
5	80	
6	20	
7	60	
8	25	
9	20	
10	25	
11	20	
12	20	
13	30	
Total:	410	

1. (20 points) What is an algorithm? Define. Explain. Use examples.

---

2. (50 points) Problem solving paradigms.

   - **(10 points)** List the 4 problem solving paradigms.
   - **(40 points)** Give 2 example algorithms that fit into each paradigm.

Note:

Full credit for this question:

- Examples clearly explained
- Why they fit into particular paradigm and not another.
- Possible counter example either with another algorithm or paradigm.

Bad Example: "Brute force" or "complete search" means you look at every item in a data structure. A "linear" search is an example because it looks at every item in a list.

Better Example: Look at definition on page 70 in CP3 to get a proper definition of brute force. "Linear Search" of an unordered list of integers fits into this category since we cannot terminate the search and must (at worst case) look at every item to determine if a value is present. Alternatively, if the list were sorted we could use a _____ (you look it up) approach and search more efficiently.

---

3. (20 points) List the complexities from fastest to slowest.

Complexity Choices

$O(n!)$ $O(2^n)$ $O(1)$ $O(n\ lg\ n)$ $O(n^2)$ $O(n^n)$ $O(n)$ $O(log\ n)$ $O(n^n)$

Note:

For another **10 points** show the values for each complexity after plugging in 1, 10 ,100, 1000 or write "undefined" if the value gets to large to handle easily.

---

4. (20 points) You are attending a party with n other people. Each other person i arrives at the party at some time s_i and leaves the party at some time t_i (where $s_i < t_i$). Once a person leaves the party, they do not return. Additionally, each person i has some coolness c_i. At all times during the party, you choose to talk to the coolest person currently at the party. (All coolness values are distinct.) If you are talking to someone, and someone else cooler arrives at the party, you leave your current conversation partner and talk to the new person. If the person you are talking to leaves the party, you go talk to the coolest person remaining at the party. (This might or might not be a person with whom you have already talked.) You are the first to arrive at the party and the last to leave. Additionally, you are the most popular person at the party, so everyone wants to talk with you.

   Describe a data structure which allows you to decide in $O(1)$ time to whom you should talk at any moment. You should be able to update this data structure in $O(logn)$ time when someone arrives or leaves.

---

5. (40 points) Spanning Trees.
   (a) (10 points) What is a spanning tree?
   (b) (10 points) What are they used for?
   (c) (20 points) What algorithms exist to find spanning trees within a graph? Do they they have the same complexities? Does each algorithm work better for certain types of graphs?

---

6. (20 points) Given the following list of values:

   $[1, 1, 2, 2, 2, 3, 3, 4, 4, 4, 4, 4, 5, 5, 5, 6, 6, 7, 7, 7, 7, 8, 8, 8, 8, 8, 8, 8, 9, 9, 9]$

   What algorithm paradigm would you use to find the number of occurrences of a given number in the fastest way possible. Use any method you want. You may even combine paradigms if you want.

   Example:
   Target: 7
   Output: 4

```
1
2 int arr[] = {1,1,2,2,2,3,3,4,4,4,4,4,5,5,5,6,6,7,7,7,7,8,8,8,8,8,8,8,9,9,9}
3 int x = find_occurences(arr,7)
4 cout << x << endl;
5 // writes out 4
6
```

Note:

Make sure you explain in detail what you are doing. Remember you are describing your algorithm for solving the problem. Anyone reading your answer should be able to re-create your steps and solve the problem themselves.

---

7. (30 points) We didn't discuss sorting algorithms in depth during class. The majority of sorting algorithms perform in a couple of similar categories which really do not effect us when sorting small chunks of data. But it is good to know about the different approaches to sorting. So for this question I want you to explain to me:

   (a) (10 points) What categories of sorting algorithms are there?
   (b) (5 points) That majority of sorting algorithms can be categorized in with run times of O(_____) or O(_____)
   (c) (15 points) What is the best performance you can expect from a sorting algorithm? Give examples and be thorough.

8. (25 points) Again, here is another "topic" we didn't discuss in class. Why? We tend to focus on problems that we can solve using algorithms that we can implement. These are the majority of problems: solvable in a reasonable amount of time with a reasonable algorithm. Reasonable = "P" or "polynomial" time. However, there is a category of problems that are not solvable in a reasonable amount of time (reasonable is relative of course). We categorize those problems differently. They are:

   - P
   - NP
   - NP Hard
   - NP Complete

   Each of these problem categories get progressively harder. Where NP Complete problems are basically problems that could take **years** to solve and P problems take **seconds** or **minutes** to solve.

   - Discuss the possibility of converting NP to P, is it possible?

   - What are the ramifications of converting a problem that is categorized as NP into a solution in P (aka making a Non-Polynomial solution solvable in Polynomial time) (this question should give you a hint that its currently not possible).
   - Also, discuss approximation algorithms and how they fit into the big picture for solving very hard problems.

---

9. (20 points) Do you notice any relationship between problem solving paradigms and P vs NP?

   Note:

   - Be careful with the next 3 questions.
   - Try to tie in each answer with a paradigm, mention complexity, and also discuss if it's an optimal solution.

---

10. (25 points) Assume you are part of a relief effort in a third world nation. You need to drop medical supplies to as many locals as possible from your AC-130 aircraft supplied to you by some participating nation. Your goal is to leave your home base, visit as many villages as possible by air, drop your supplies to each, and return to home base.

    - Describe an algorithm to help you achieve this goal.
    - Is there an existing algorithm to help you solve this problem?
    - Will the algorithm provide an optimal solution?

---

11. (20 points) You are given the task of writing a class to implement "undo" / "redo" for a popular word processor.

    - What data structure do you choose?

- Describe your algorithm for both undo and redo. Do they use the same data structure?
- Do you use static memory allocation or dynamic memory allocation?

---

12. (20 points) You are in charge of campus tours for the school year 2020-2021. Forget about covid19 and concentrate on your tours. There are a minimum set of places that each family needs to visit in varying locations on campus. They are all over campus and can be reached from multiple paths, and in any order. Ideally you would start in one place, visit every location, and end up where you started.

    - How do you solve this problem?
    - Do you use a data structure an algorithm or both?
    - Can your solution satisfy all involved? Meaning will it be efficient and not waste time?

---

13. (30 points) You have been tasked to come up with a schedule of courses for Spring 2021 semester. You are given all the necessary course requirements (classes that need to be scheduled) and the list of available rooms on campus.

    - What algorithms will you use to perform the scheduling?
    - What problem solving paradigm is your algorithm in?
    - Can you com up with a perfect solution?
