CSE 373: 23 Spring Practice Final

Name:	UW Email:	@uw.edu

Instructions

- The allotted time is **50** minutes. Please do not turn the page until the staff says to do so.
- This is an open-book and open-notes exam. You are NOT permitted to access electronic devices including calculators.
- Read the directions carefully, especially for problems that require you to show work or provide an explanation.
- We can only give partial credit for work that you've written down.
- Unless otherwise noted, every time we ask for algorithm runtime, it must be **simplified and tight**.
- If you run out of room on a page, indicate where the answer continues. Try to avoid writing on the very edges of the pages: we scan your exams and edges often get cropped off.

Advice

- If you feel like you're stuck on a problem, you may want to skip it and come back at the end if you have time.
- Relax and take a few deep breaths. You got this :-)

Questions	Max Points
1. Sorting	0
2. Graphs	0
Total	0

Resubmission Details

- This exam will be graded out of 100 points. If you are not satisfied with your grade, you will be given the opportunity to resubmit it online and earn up to 50% of the missed points back.
- For example, a student scoring 80/100 points may receive up to 90/100 points on the resubmission.

1 Sorting

Simon's little brother, 🅱️imon, just learned about binary search trees in CSE 123. After realizing how useful they were, 🅱️imon came up with his own sorting algorithm! He creatively named it `BSTSort`. Here's the pseudocode:

```
BSTSort(array):
    initialize an empty binary search tree, B
    for each element e in array:
        add e to B
    perform an inorder traversal of B (and print out/return
    the elements in this order)
```

🅱️imon hasn't taken 373 yet, so he doesn't know any of the fancy data structures you learned in class or how to analyze sorting algorithms. 🅱️imon knows that you're a CS expert, so he's asked you for your input.

What is the best-case runtime for this algorithm in terms of the size of the array, `n`? Provide a simplified and tight Big-Oh bound, then describe or provide an input array that would result in best-case performance.

What is the worst-case runtime for this algorithm in terms of `n`? Provide a simplified and tight Big-Oh bound, then describe or provide an input array that would result in worst-case performance.

Recall that by the definition of a Binary Search Tree, all elements to the right of a node are greater than or equal to the node itself.

Is the sort stable? No justification required.

Is the sort in-place? If not, provide a simplified and tight Big-Oh bound for the extra space complexity.

Finally, what is one modification you could make to the algorithm so that the worst case runtime matches that of the best case? For this part, assume no duplicates.

2 Graphs

2.1 Final Grading

The TAs are going to meet in the secret CSE 3 building to grade the final. Before they meet, each TA is located at a different building on campus. They are all going to head out to CSE 3 at exactly the same time, but some TAs will arrive earlier than others. Each TA is familiar with campus, so they naturally take the shortest path from their starting point to CSE 3, and they all walk at an average pace. You want to determine which TA will arrive first. The path a TA takes can be broken into a series of straight-line segments of pavement, connecting one building to the next. You are given as input a map from each starting building to the name of the corresponding TA. Any additional data that could help you solve the problem, such as data about campus distances, is available to you online.

You decide to model this problem with a graph.

In one word, what should each vertex represent in the graph?

In one short sentence, what should each edge represent in the graph?

Should the edges in the graph be directed or undirected?

Select the best answer:

- ◯ Directed
- ◯ Undirected

Should the edges in the graph be weighted or unweighted?

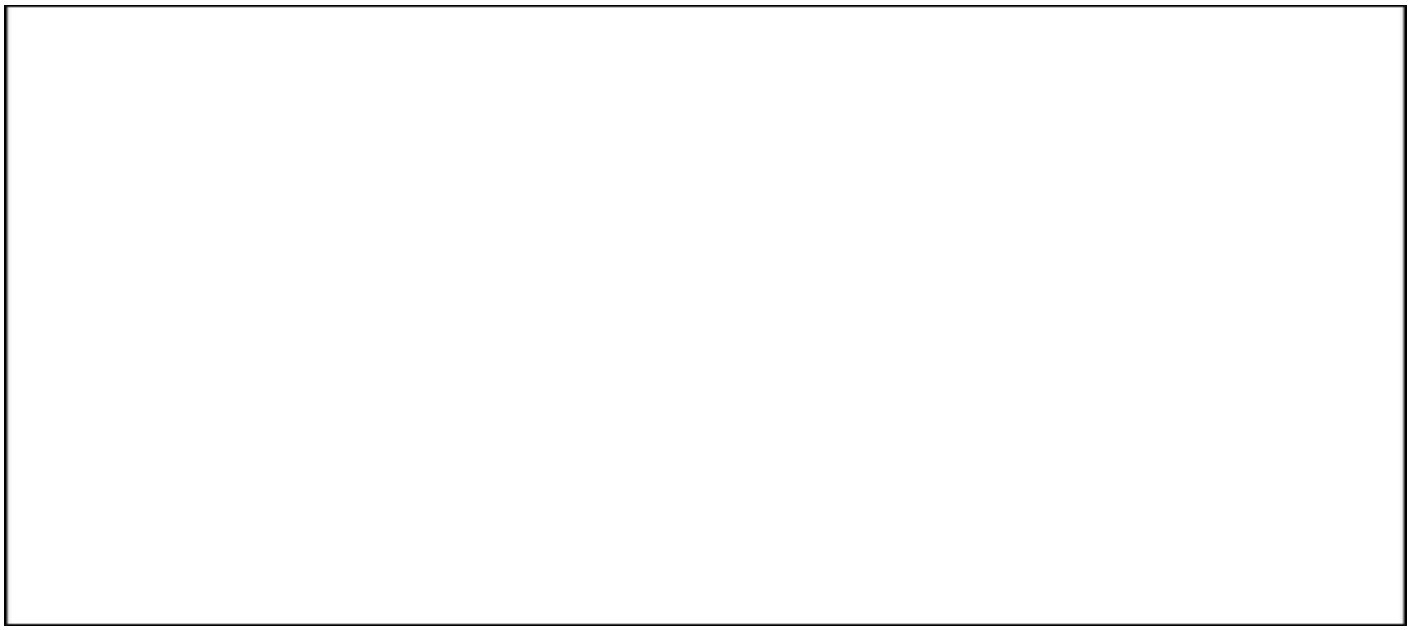
Select the best answer:

- ◯ Weighted
- ◯ Unweighted

If you answered “Weighted” to the previous question, in one phrase, what should each edge weight represent? If you answered “Unweighted” to the previous question, leave it blank.

Explain how you can efficiently determine which TA arrives first, assuming the graph above has already been constructed. You **do not** need to explain how the algorithm is implemented. For example, if you want to use breadth-first search in your answer, simply say “use BFS to achieve…”. For your algorithm to be considered “efficient”, it must call Djikstra’s algorithm

at most once.



2.2 Robots

Two robots are in a warzone, and they each want to return to their home base safely. The robots are on opposing sides, so if they get within sight of each other they will explode. Being a neutral pacifist, you want to help both robots return safely.

We can model the situation with a graph. Consider the **undirected**, **unweighted** graph $G = (V, E)$, representing the warzone. The two robots are initially located at vertices a and b. The robot at vertex a wants to get to vertex c along a path in G, and likewise the robot at b wants to get to vertex d. This is accomplished with a schedule: at each time step, the schedule specifies that one of the robots moves across a single edge, from one vertex to a neighboring vertex. At the end of the schedule, the robot from vertex a should be sitting on c, and the robot from b should be sitting on d. Your task is to design an algorithm that returns true if and only if there is a schedule so that no robot is within a distance r of each other at any time step. Distance is measured by the number of edges along the shortest path between two vertices.

You decide to solve this problem by creating a new graph, *H*. As a hint, a vertex in *H* represents the "current locations" of the two robots. This can be represented as a pair of vertices in *G*.

In one short sentence, what does each edge represent in the graph?

Are the edges in the graph directed or undirected?

Select the best answer:

- ◯ Directed
- ◯ Undirected

Are the edges in the graph weighted or unweighted?

Select the best answer:

- ◯ Weighted
- ◯ Unweighted

If you answered "Weighted" to the previous question, in one word, what does each edge weight represent? If you answered "Unweighted" to the previous question, leave it blank.

Explain how you can determine whether such a schedule exists. If you plan to use one of the graph algorithms you've learned in class, you do not need to explain how the algorithm is implemented. For example, if you want to use breadth-first search in your answer, simply say "use BFS to achieve…".