# Design & Analysis of Algorithms
## Midterm
## Info
### Date: Wednesday, March 13, 2024, 9:15 am – 10:30 am

- This is a closed book exam; only 1 sheet of notes (handwritten) is permitted.
- You will have 75 minutes for this exam.
- It will consist of 5 problems worth 10 points each, plus a problem 6 for extra credit.
- The extra credit will be recorded separately, so make sure you have answered all questions from the first 5 problems before moving on to problem 6.
- All questions will be taken from the following chapters (for list of sections see the web page of the class http://www.cs.kent.edu/~dragan/CS4-56101-D&AofAlgS24.html).

1. **Algorithm Analysis**
2. **Basic Data Structures**
3. **Search Trees and Skip Lists**
4. **Sorting, Sets, and Selection**

**Possible Questions:** A problem may consist of few questions. You should expect a question of the form: work through some algorithm on a given input, showing intermediate results. I'll pick the algorithm at random, because I want you to learn them all. For some more "complex" algorithms (e.g. Red-Black tree manipulation), I may ask you to do only one operation. You should also expect a series of short-answer questions. Although the answers are short, some of the questions may require some thought. Don't worry about skipping a question (or guessing an answer) if you don't see the answer right away.

The other problems may involve designing and analyzing simple algorithms. These are much more unpredictable. I try to ask at least a couple of questions that involve a very simple modification to a homework problem, so be sure you understand the solutions to all the homework problems. Other questions usually involve a slight modification to a problem that we have worked on in class.

Finally, I have one challenging problem. Don't be upset if you cannot solve this problem. (But at least be sure to read it, since sometimes the problem is not as challenging as I think it is, and there may be a very simple solution.) If you cannot see how to solve the problem, feel free to write down your wild ideas or observations. You might get some partial credit if your idea is good.

When it comes to grading, I tend to give more credit for ``insight'' than for ``memorization''. For example, if a problem involves inorder traversal of a binary tree, you will get little credit for just writing down the algorithm. (You could just copy that out of your cheat sheet.) However, if you can say something intelligent about the mathematical structure of the particular problem, or about why some approach is not useful, you may get some partial credit (depending on how much your observation indicates about your understanding of the problem).