

Data Structures – DAST

Exam guidelines – Spring 2004

Last revised: May 30, 2004.

Exam date and location: Wed. June 24.

1. Exam guidelines

- the exam will last 2.5 hours.
- the exam is closed-book and closed-notes.
- the exam covers the material detailed in the exam syllabus.
- no calculators or computers are allowed – they are not necessary.
- a list of standard formulas will be provided in the exam.

2. Exam structure

- the exam is in two parts, with a total of 4 questions of equal weight, each worth 25 points.
- the exam allows choice.
- Part 1: choose 6 out of 8 short questions. Must be answered by all.
 - o YES/NO or multiple choice answer + short explanation (up to 5 lines).
 - o each question is worth 4 credits and will be graded as follows:
 - right answer, right explanation → 4 points
 - right answer, incomplete explanation → 2—3 points
 - right answer, wrong or no explanation → 1 point
 - wrong answer, regardless of explanation → 0 points
 - o one point is free for all (6 questions x 4 points = 24 points + 1 point free).
 - o if there are more than six questions answered, only the first six will be considered.
- Part 2: choose 3 out of 4 long questions.
 - o each question consists of 3-4 sub-questions.
 - o the level of difficulty of each sub-question is indicated by its number of points.
 - o solving part of one question and part of another is not allowed.
 - o if there are more than three questions answered, only the first three will be considered.
- the first part and the three questions in the second part are designed to be solved in about 30mins each, for a total of 2 hours.
- the additional 30mins are for reading carefully ALL the questions, for choosing which ones to answer, and for carefully writing your answers.

3. Exam presentation

- you must fill your answers on the exam sheet (paper) itself.
- no additional space is allowed. Your answers must fit in the allocated space.
- you are encouraged to use the exam booklets as scrap paper, and write on the exam your answers only. The exam booklets will be collected at the end of the exam but will not be considered in the exam evaluation.

4. Style of questions and expected answers

- answer precisely and briefly.
- drawings are an excellent tool for making yourself clear. They are strongly encouraged. However, drawings with out explanations are not considered proofs.
- the level of detail and rigor expected is determined by the question phrasing as follows:

- Arguments, properties, complexity
 - o Prove/disprove: provide a rigorous mathematical proof similar to the ones presented in class, or give a precise counter-example.
 - o Proof methods: direct, induction, contradiction, counter-example.
 - o Show: provide a detailed argument based on results from proofs seen in class.
- Algorithms
 - o Pseudo-code: provide pseudo-code at the level of the detail seen in class.
 - o Describe: explain how the algorithm works in terms of algorithms and data structures seen in class. Explain what is different from the method/data structure seen in class.
- Difficulty
 - o the level of difficulty is less or about that of questions in the theoretical homeworks.
 - o some questions require developing new algorithms and data structures. Others require combinations or variations of algorithms and data structures presented in class.
 - o you will be asked to develop algorithms for given problems based on variations of the algorithms seen in class and in the tirgul.

5. Exam grade

- results will be published by Sunday July 11.
- you will have 2 weeks to present an IRUR.
- your final grade is 65% exam, 25% programming and 10% theoretical homework.
- the course average grade is expected to be 82—84.

6. Preparing for the exam

- read carefully all the material in the syllabus and make sure you understand it.
- do not attempt to memorize pseudo-code or proofs. Instead, understand them and ask yourself what are the main ideas behind the proof/code.
- make sure you understand the assumptions made in each case.
- compare different algorithms and data structures and make sure you know their relative advantages and disadvantages.
- make sure you know how to define the basic concepts (complexity functions, comparison sorting, minimum spanning tree, etc) very precisely.
- make sure you know the worst and average case space and time complexity of the algorithms seen in class.
- practice on the book exercises and on previous exams.
- studying and solving problems in groups is encouraged!

Sample questions

Part 1 -- Question 1 (short questions –25%)

1. The average-case complexity of BubbleSort is $O(n \lg n)$.

Answer (1%): FALSE.

Justification (3%):

Bubble sort consists of two loops, the outer loop which always iterates from $i=1$ to $i=n-1$. The inner loop goes down from n to i and swaps between contiguous elements if necessary. Therefore, regardless of the input, the Sum from 1 to n , indexed by i , over $n-i$ is then $O(n^2)$ and not $O(n \lg n)$.

Part 2 -- Question 2 (long question –25%)

Maximum spanning trees. Let $G=(V,E)$ be a connected, directed graph with weighted edges. Define the maximum spanning tree of G , $maxST=(V,E')$ as the tree included in G whose edge weight sum is maximum.

1 (4%). Are maximum spanning trees unique? Prove or disprove.

2 (6%). Let U be a subset of V , and e an edge with the largest weight connecting a vertex from U and a vertex from $V-U$. Prove that there exists a maximum spanning tree that includes e .

3 (10%). Describe an algorithm (pseudo-code) and a data structure for finding a maximum spanning tree.

4 (5%). Analyze the space and time complexity of your algorithm.