Exam 26 October 2012, 8:00–13:00, Sparta:A–B

EDAN55 Advanced Algorithms

preliminary alpha draft

The exam consists of 4 large questions; each consisting of a number of smaller subquestions.

1. The exam is "open book," so you can bring whatever material you want, including textbooks, a dictionary, and your own course notes.

2. You can bring an electronic calculator, even though I don't see how that is useful.

3. We try to minimise the dependencies among subquestions. In particular, you can solve them in any order and are free to *use* the result of subquestion x to answer subquestion y, even if you didn't answer x.

4. Scoring: Answering "I don't know" (and nothing else) scores $\frac{1}{4}$ of a subquestion's points. An empty or wrong answer scores 0 points.

5. You can answer in Swedish or English.

Some tips:

1. Shorter is better.

2. An example is better than a failed attempt at explaining something in general.

3. Drawings, pseudocode, and formulas are good. "Wall of text" is bad.

4. Admit ignorance.

5. Be tidy.

Good luck!

Question 1, Approximation

The [redacted] is defined as follows. Given an undirected graph $G = (V, E)$, find a [redacted] such that [redacted] is maximised. Formally, a [redacted] Define the [redacted]

[redacted]

We want to find a [redacted] of maximum [redacted]

►*1a* (1 pt.) Find a maximum [redacted] the graph in fig. 1.[1]

Consider the following simple algorithm for [redacted]. Let $V = \{v_1, \ldots, v_n\}$ and for [redacted]

1. Set [redacted]
2. For [redacted]

►*1b* (1 pt.) Run the algorithm on the graph in fig. 1 and give the result. [2]

►*1c* (1 pt.) Give an example where the algorithm finds a [redacted]

►*1d* (3 pts.) Show that the algorithm is guaranteed to [redacted] [4]

►*1e* (2 pts.) Prove that there cannot be an algorithm for [redacted] that approximates the optimum solution within [redacted] unless [redacted] You can freely use that [redacted] is [redacted]

►*1f* (1 pt.) Modify the algorithm and analysis for the [redacted] problem, where [redacted]

►*1g* (1 pt.) Modify the algorithm and analysis for the [redacted] problem, where t[redacted]

[1] Your answer is a drawing showing [redacted] an integer [redacted]

[2] Your answer is t[redacted]

[3] Your answer is a concrete graph, an optimum solution to that instance and the solution found by the algorithm.

[4] Your answer is a short proof. It includes a lower bound on the solution found by the algorithm and an upper bound on the optimum solution.

[5] You answer is a short proof.

[6] You answer contains an algorithm and a brief analysis of its approximation guarantee; the most important part is to state the resulting approximation factor.

[7] You answer contains an algorithm and a brief analysis of its approximation guarantee; the most important part is to state the resulting approximation factor

Question 2, Parameterized Analysis

[redacted] [redacted], such as in figure ??. We consider the problem of [redacted] Formally,

Name: [redacted]

Input: A set [redacted] Integer k.

Output: A set of k [redacted]

[redacted]

▶ *1a* (1 pt.) Find a [redacted][8]

▶ *1a* (2 pts.) Write a simple exhaustive search (or "brute force") algorithm and give its running time in terms of n and k.[9]

We will write a better algorithm. The central observation is that if [redacted]

▶ *1a* (1 pt.) Why is this true?

▶ *1a* (1 pt.) Give a counterexample that shows that this is not true for [redacted]

▶ *1a* (4 pts.) Write an algorithm based on the above observation, briefly argue for its correctness, and state its running time. (The running time must have the form $f($[redacted]$)n^{O(1)}$ for some function f.)

[8] Your answer is a drawing showing [redacted]

[9] Your answer is some lines of pseudocode and a running time estimate using asymptotic notation.

Question 3, Exponential Time Algorithms

We consider [redacted]

Name: [redacted]

Input: A [redacted]

Output: A smallest [redacted]

[redacted]

$j \in J$

[redacted]

For instance, the graph in fig. ?? contains a [redacted]

►*4a* (1 pt.) Find a [redacted] for the instance in fig. ??.

►*4a* (2 pt.) Explain very briefly how [redacted] can be solved using exhaustive search ("brute force") and state the resulting running time.

►*4a* (1 pt.) Explain very briefly why [redacted]

►*4a* (4 pts.) Construct a branching ("decrease-and-conquer") algorithm for [redacted] You running time must be better than 2^n. Be precise about which rules you use; for example by writing the algorithm in some form of pseudocode. Briefly argue why each rule is valid. Give a recurrence relation for the running time of the resulting algorithm; read the solution to your recurrence off Table 1. *Hint:* [redacted]

	1	2	3	4	5
1	2	1.62	1.47	1.39	1.33
2		1.42	1.33	1.28	1.24
3			1.26	1.23	1.20
4				1.19	1.17
5					1.15

Table 1: Running times for decrease-and-conquer recurrences of the form $f(N) \leq f(N-a) + f(N-b)$. For example, the Fibonacci reccurence $F(N) = F(N-1) + F(N-2)$ satisfies $F(N) = O(1.62^N)$.

Question 4, Randomized Algorithms

We consider the [redacted]

Name: [redacted]

Input: A simple, undirected graph $G = (V, E)$ with n vertices and m edges.

Output: A [redacted] (or both).

Consider the following randomized algorithm for this problem:

1. For every [redacted] pick [redacted] at random from [redacted]
2. Let [redacted]

   where $N(v)$ denotes the neighbours of v.

▶*4a* (1 pt.) Run the algorithm on the graph in fig. ??. [redacted]

▶*4b* (1 pt.) [redacted] Find the probability that [redacted] [11]

▶*4c* (1 pt.) Consider running the algorithm on [redacted] Are the events [redacted] and [redacted] independent? Why or why not? [12]

▶*4d* (1 pt.) What is the probability that [redacted]

▶*4e* (2 pt.) Assume that G is d-regular (i.e., every vertex as degree exactly d). Find [redacted] expected size of [redacted]

▶*4f* (1 pt.) Assume G is the n-cycle. (That is, $E = \{\,\{i, i+1\} : 1 \leq i < n\,\} \cup \{n, 1\}$.) Find the expected [redacted]

▶*4g* (1 pt.) Assume G is the n-star. (That is, $E = \{\,\{1, i\} : 2 \leq i \leq n\,\}$.) What is the probability that [redacted] [16]

[10] Your answer is a drawing [redacted]

[11] Your answer is an expression and an argument for it.

[12] Your answer is the word "yes" or the word "no", followed by an argument.

[13] This is not meant to be a trick question. But if you think your answer is weird, it's probably correct.

[14] Your answer is an expression and an argument for it.

[15] Your answer is an expression and an argument for it.

[16] Your answer is an expression and an argument for it.

►*4g* (1 pt.) Assume G is the n-star. (That is, $E = \{\{1, i\} : 2 \leq i \leq n\}$.) Calculate th[redacted] probability of [redacted] [17]

[17] Your answer is an expression and an argument for it.