

PhD Comprehensive Exam: Math and Algorithms

Spring 2000

The exam includes eight problems;
the length of the exam is three hours.

Problem 1

Consider the problem of finding the k th smallest element of an array, that is, the element that would occupy the k th position after sorting the array. For example, if the array is $\langle 6, 4, 8, 2, 10, 0 \rangle$ and $k = 3$, then the k th smallest element is 4, since it is the third element in the sorted array $\langle 0, 2, 4, 6, 8, 10 \rangle$.

Give an algorithm for finding the k th smallest element. The average-case time of your algorithm should be *significantly smaller* than the time of standard sorting procedures. Thus, sorting the whole array and then returning the k th element is not an appropriate solution.

Problem 2

Consider two recurrences:

$$\begin{aligned}A_0 &= 0 \\ A_n &= A_{n-1} + n \quad (\text{where } n \geq 1)\end{aligned}$$

$$\begin{aligned}B_0 &= 0 \\ B_n &= B_{n-1} + A_n \quad (\text{where } n \geq 1)\end{aligned}$$

Give a formula for B_n in terms of n , without the use of recurrence. Note that you should provide an exact formula, rather than a Θ -notation.

Problem 3

Write an algorithm `COMPONENTS(G)` that identifies connected components of an undirected graph G ; make your algorithm as efficient as possible.

Problem 4

Suppose that e is the minimum-weight edge of an undirected weighted graph, and that all other edges have *strictly* larger weights. Prove that every minimum spanning tree of this graph includes the edge e .

Problem 5

King Arthur once invited a number of knights to his castle, where they stayed for several days. Each evening, the king and his guests dined at the Round Table; in total, there were *twenty-three* people at the table. According to the king's decree, they took different seats on different evenings, and *no two people sat next to each other more than once*. When the knights could no longer satisfy this decree, they left the king's castle. What is the maximal number of days they could stay in the castle?

Note: The setting of this problem is very similar to Problem 4 of the previous comprehensive exam; however, the question is different, and it requires a completely different solution.

Problem 6

Suppose that you have a large file of English words; for convenience, assume that all letters in the file are lower-case. One of the words occurs once in the file, and all other words occur twice. Describe an efficient procedure for identifying the word that occurs once. For example, suppose that the file includes the following nine words:

hour	day	week
week	year	hour
month	day	month

Then, the procedure must return the word **year**. Your algorithm must run in *linear* time and use *constant* amount of memory.

Problem 7

The *chromatic number* of an undirected graph is the minimal number of colors required for painting all vertices, in such a way that adjacent vertices have different colors.

(a) A graph is called *planar* if we can draw it in the plane, in such a way that its edges do not intersect; note that edges in this drawing do not have to be straight, that is, we may draw curvilinear edges. Give an example of a planar graph for which the chromatic number is 4.

(b) The *degree* of a vertex in an undirected graph is the number of adjacent vertices. We denote the maximal vertex degree in a given graph by d . Prove that the chromatic number of a graph is no larger than $d + 1$, and give an example of a graph for which the chromatic number is strictly smaller than $d + 1$.

(c) Describe a fast algorithm for painting the vertices of a graph, using at most $d + 1$ colors.

Problem 8

In the *Three-Coloring Problem*, we need to determine whether the vertices of a given undirected graph can be painted with three colors, in such a way that no two adjacent vertices have the same color. Similarly, in the *Four-Coloring Problem*, we need to determine whether the vertices can be painted with four colors.

The Three-Coloring Problem is known to be NP-complete. Use this fact to prove that the Four-Coloring Problem is also NP-complete.