

You have 120 minutes to answer four of these five questions.

Write your answers in the separate answer booklet.

1. Multiple Choice.

Each of the questions on this page has one of the following five answers:

A: $\Theta(1)$ B: $\Theta(\log n)$ C: $\Theta(n)$ D: $\Theta(n \log n)$ E: $\Theta(n^2)$

Choose the correct answer for each question. Each correct answer is worth +1 point; each incorrect answer is worth $-\frac{1}{2}$ point; each "I don't know" is worth $+\frac{1}{4}$ point. Your score will be rounded to the nearest *non-negative* integer. You do *not* need to justify your answers; just write the correct letter in the box.

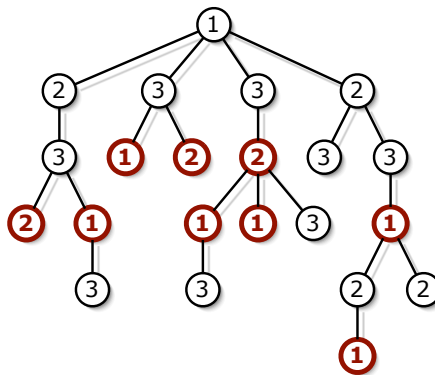
- (a) What is $\frac{5}{n} + \frac{n}{5}$?
- (b) What is $\sum_{i=1}^n \frac{n}{i}$?
- (c) What is $\sum_{i=1}^n \frac{i}{n}$?
- (d) How many bits are required to represent the n th Fibonacci number in binary?
- (e) What is the solution to the recurrence $T(n) = 2T(n/4) + \Theta(n)$?
- (f) What is the solution to the recurrence $T(n) = 16T(n/4) + \Theta(n)$?
- (g) What is the solution to the recurrence $T(n) = T(n-1) + 1/n^2$?
- (h) What is the worst-case time to search for an item in a binary search tree?
- (i) What is the worst-case running time of quicksort?
- (j) What is the running time of the fastest possible algorithm to solve Sudoku puzzles? A Sudoku puzzle consists of a 9×9 grid of squares, partitioned into nine 3×3 sub-grids; some of the squares contain digits between 1 and 9. The goal of the puzzle is to enter digits into the blank squares, so that each digit between 1 and 9 appears exactly once in each row, each column, and each 3×3 sub-grid. The initial conditions guarantee that the solution is unique.

2								4
	7		5					
				1		9		
6		4			2			
	8							5
			9			3		7
		1		4				
					3		8	
	5							6

A Sudoku puzzle. Don't try to solve this during the exam!

2. Oh, no! You have been appointed as the gift czar for Giggle, Inc.'s annual mandatory holiday party! The president of the company, who is certifiably insane, has declared that *every* Giggle employee must receive one of three gifts: (1) an all-expenses-paid six-week vacation anywhere in the world, (2) an all-the-pancakes-you-can-eat breakfast for two at Jumping Jack Flash's Flapjack Stack Shack, or (3) a burning paper bag full of dog poop. Corporate regulations prohibit any employee from receiving the same gift as his/her direct supervisor. Any employee who receives a better gift than his/her direct supervisor will almost certainly be fired in a fit of jealousy. How do you decide what gifts everyone gets if you want to minimize the number of people that get fired?

More formally, suppose you are given a rooted tree T , representing the company hierarchy. You want to label each node in T with an integer 1, 2, or 3, such that every node has a different label from its parent. The *cost* of an labeling is the number of nodes that have smaller labels than their parents. Describe and analyze an algorithm to compute the minimum cost of any labeling of the given tree T . (Your algorithm does *not* have to compute the actual best labeling—just its cost.)



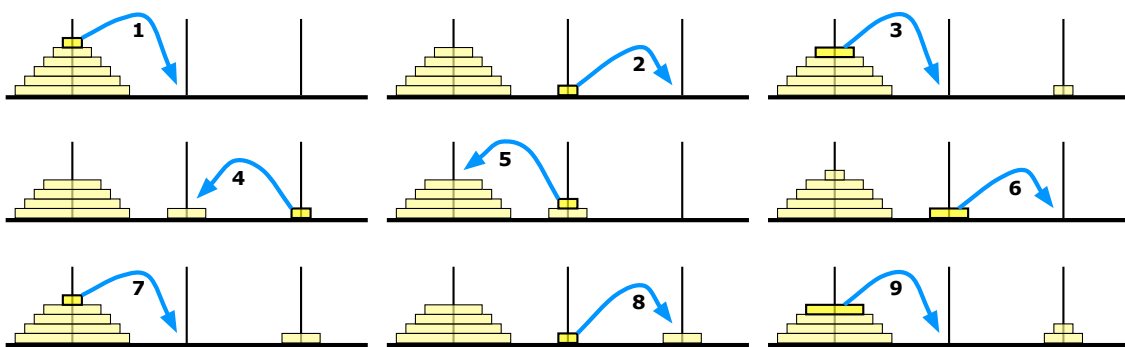
A tree labeling with cost 9. Bold nodes have smaller labels than their parents. This is *not* the optimal labeling for this tree.

3. Suppose you are given an array $A[1..n]$ of n distinct integers, sorted in increasing order. Describe and analyze an algorithm to determine whether there is an index i such that $A[i] = i$, in $o(n)$ time. [Hint: Yes, that's *little*-oh of n . What can you say about the sequence $A[i] - i$?
4. Describe and analyze a *polynomial-time* algorithm to compute the length of the longest common subsequence of two strings $A[1..m]$ and $B[1..n]$. For example, given the strings 'DYNAMIC' and 'PROGRAMMING', your algorithm would return the number 3, because the longest common subsequence of those two strings is 'AMI'. You must give a complete, self-contained solution; don't just refer to HW1.

5. Recall that the Tower of Hanoi puzzle consists of three pegs and n disks of different sizes. Initially, all the disks are on one peg, stacked in order by size, with the largest disk on the bottom and the smallest disk on top. In a single move, you can transfer the highest disk on any peg to a different peg, except that you may never place a larger disk on top of a smaller one. The goal is to move all the disks onto one other peg.

Now suppose the pegs are arranged in a row, and you are forbidden to transfer a disk directly between the left and right pegs in a single move; every move must involve the middle peg. How many moves suffice to transfer all n disks from the left peg to the right peg under this restriction? **Prove your answer is correct.**

For full credit, give an *exact* upper bound. A correct upper bound using $O(\cdot)$ notation (with a proof of correctness) is worth 7 points.



The first nine moves in a restricted Towers of Hanoi solution.