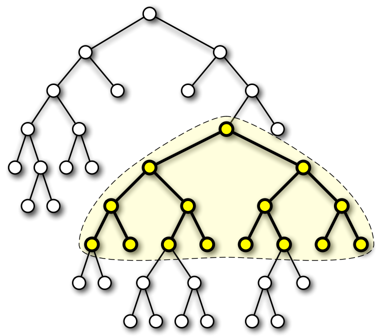
You have 90 minutes to answer four of these questions.
Write your answers in the separate answer booklet.
You may take the question sheet with you when you leave.

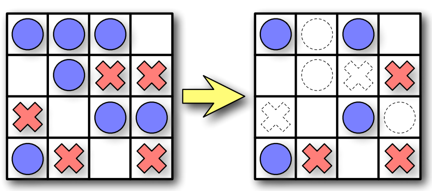
1. Recall that a binary tree is *complete* if every internal node has two children and every leaf has the same depth. An *internal subtree* of a binary tree is a connected subgraph, consisting of a node and some (possibly all or none) of its descendants.

   Describe and analyze an algorithm that computes the depth of the *largest complete internal subtree* of a given n-node binary tree. For full credit, your algorithm should run in $O(n)$ time.

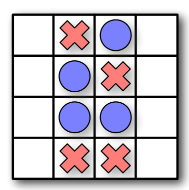


The largest complete internal subtree in this binary tree has depth 3.

2. Consider the following solitaire game. The puzzle consists of an $n \times m$ grid of squares, where each square may be empty, occupied by a red stone, or occupied by a blue stone. The goal of the puzzle is to remove some of the given stones so that the remaining stones satisfy two conditions: (1) every row contains at least one stone, and (2) no column contains stones of both colors. For some initial configurations of stones, reaching this goal is impossible.



A solvable puzzle and one of its many solutions.



An unsolvable puzzle.

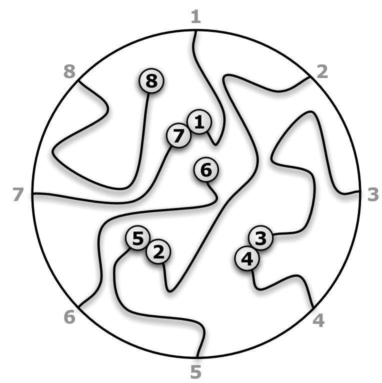
   Prove that it is NP-hard to determine, given an initial configuration of red and blue stones, whether the puzzle can be solved.

3. Suppose you are given two sorted arrays $A[1..n]$ and $B[1..n]$ and an integer k. Describe and analyze an algorithm to find the kth largest element in the union of A and B in $O(\log n)$ time. For example, given the input

$$A[1..8] = [0, 1, 6, 9, 12, 13, 18, 21], \qquad B[1..8] = [2, 4, 5, 8, 14, 17, 19, 20], \qquad k = 10,$$

   your algorithm should return 13. You can assume that the arrays contain no duplicates. *[Hint: What can you learn from comparing one element of A to one element of B?]*

4. Every year, as part of its annual meeting, the Antarctican Snail Lovers of Upper Glacierville hold a Round Table Mating Race. Several high-quality breeding snails are placed at the edge of a round table. The snails are numbered in order around the table from 1 to n. During the race, each snail wanders around the table, leaving a trail of slime behind it. The snails have been specially trained never to fall off the edge of the table or to cross a slime trail, even their own. If two snails meet, they are declared a breeding pair, removed from the table, and whisked away to a romantic hole in the ground to make little baby snails. Note that some snails may never find a mate, even if the race goes on forever.



The end of a typical Antarctican SLUG race. Snails 6 and 8 never find mates.
The organizers must pay $M[3,4] + M[2,5] + M[1,7]$.

For every pair of snails, the Antarctican SLUG race organizers have posted a monetary reward, to be paid to the owners if that pair of snails meets during the Mating Race. Specifically, there is a two-dimensional array $M[1\,..\,n, 1\,..\,n]$ posted on the wall behind the Round Table, where $M[i,j] = M[j,i]$ is the reward to be paid if snails i and j meet.

Describe and analyze an algorithm to compute the maximum total reward that the organizers could be forced to pay, given the array M as input.

5. SUBSETSUM and PARTITION are two closely-related NP-hard problems.

   - SUBSETSUM: Given a set X of positive integers and an integer t, determine whether there is a subset of X whose elements sum to t.
   - PARTITION: Given a set X of positive integers, determine whether X can be partitioned into two subsets whose elements sum to the same value.

   (a) Describe a polynomial-time reduction from SUBSETSUM to PARTITION.

   (b) Describe a polynomial-time reduction from PARTITION to SUBSETSUM.

   Don't forget to ***prove*** that your reductions are correct.