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

1. Each of these ten questions 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 $-1/2$ point; each "I don't know" is worth $+1/4$ point. Your score will be rounded to the nearest *non-negative* integer.

   (a) What is $\displaystyle\sum_{i=1}^{n} \frac{n}{i}$?

   (b) What is $\displaystyle\sqrt{\sum_{i=1}^{n} i}$?

   (c) How many digits are required to write 3^n in decimal?

   (d) What is the solution to the recurrence $D(n) = D(n/\pi) + \sqrt{2}$?

   (e) What is the solution to the recurrence $E(n) = E(n - \sqrt{2}) + \pi$?

   (f) What is the solution to the recurrence $F(n) = 4F(n/2) + 3n$?

   (g) What is the worst-case time to search for an item in a binary search tree?

   (h) What is the worst-case running time of quicksort?

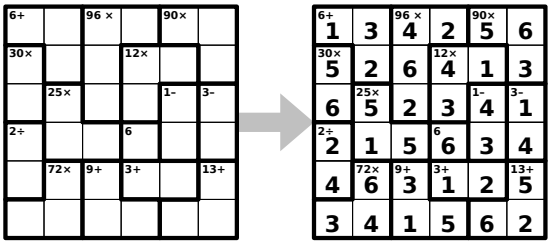
   (i) Let $H[1..n, 1..n]$ be a fixed array of numbers. Consider the following recursive function:

   $$Glub(i,j) = \begin{cases} 0 & \text{if } i = 0 \\ \infty & \text{if } i > n \text{ or } j = 0 \\ \max\{Glub(i-1,j),\ H[i,j] + Glub(i+1,j-1)\} & \text{otherwise} \end{cases}$$

   How long does it take to compute $Glub(n,n)$ using dynamic programming?

   (j) What is the running time of the fastest possible algorithm to solve KenKen puzzles?

   A KenKen puzzle is a 6×6 grid, divided into regions called *cages*. Each cage is labeled with a numerical *value* and an arithmetic *operation*: $+$, $-$, $\times$, or $\div$. (The operation can be omitted if the cage consists of a single cell.) The goal is to place an integer between 1 and 6 in each grid cell, so that no number appears twice in any row or column, and the numbers inside each cage can be combined using *only* that cage's operation to obtain that cage's value. The solution is guaranteed to be unique.

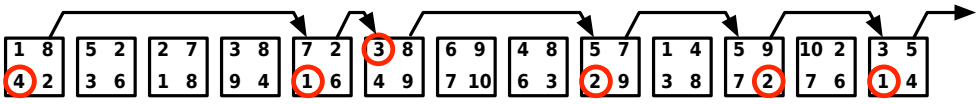


A Kenken puzzle and its solution

2. (a) Suppose $A[1..n]$ is an array of n distinct integers, sorted so that $A[1] < A[2] < \cdots < A[n]$. Each integer $A[i]$ could be positive, negative, or zero. Describe an efficient algorithm that either computes an index i such that $A[i] = i$ or correctly reports that no such index exists. An algorithm that runs in $\Theta(n)$ time is worth at most 3 points.

   (b) Now suppose $A[1..n]$ is a sorted array of n distinct ***positive*** integers. Describe an even faster algorithm that either computes an index i such that $A[i] = i$ or correctly reports that no such index exists. *[Hint: This is **really** easy!]*

3. *Moby Selene* is a solitaire game played on a row of n squares. Each square contains four positive integers. The player begins by placing a token on the leftmost square. On each move, the player chooses one of the numbers on the token's current square, and then moves the token that number of squares to the right. The game ends when the token moves past the rightmost square. The object of the game is to make as many moves as possible before the game ends.



A Moby Selene puzzle that allows six moves. (This is ***not*** the longest legal sequence of moves.)

   (a) ***Prove*** that the obvious greedy strategy (always choose the smallest number) does not give the largest possible number of moves for every Moby Selene puzzle.

   (b) Describe and analyze an efficient algorithm to find the largest possible number of legal moves for a given Moby Selene puzzle.

4. Consider the following algorithm for finding the largest element in an unsorted array:

```
RandomMax(A[1..n]):
  max ← ∞
  for i ← 1 to n in random order
      if A[i] > max
          max ← A[i]    (⋆)
  return max
```

   (a) In the worst case, how many times does RandomMax execute line $(\star)$?

   (b) What is the ***exact*** probability that line $(\star)$ is executed during the last iteration of the for loop?

   (c) What is the ***exact*** expected number of executions of line $(\star)$? (A correct $\Theta(\,)$ bound is worth half credit.)

5. *This question is taken directly from HBS 0.* Whenever groups of pigeons gather, they instinctively establish a *pecking order*. For any pair of pigeons, one pigeon always pecks the other, driving it away from food or potential mates. The same pair of pigeons always chooses the same pecking order, even after years of separation, no matter what other pigeons are around. Surprisingly, the overall pecking order can contain cycles—for example, pigeon A pecks pigeon B, which pecks pigeon C, which pecks pigeon A.

   Prove that any finite set of pigeons can be arranged in a row from left to right so that every pigeon pecks the pigeon immediately to its left. Pretty please.