1st tutorial

Jamie Fravel
jfravel@iuuk.mff.cuni.cz

1. Missing Number We are given an unsorted array A which contains n pairwise different numbers from set $\{0, \ldots, n\}$. One number from this set is missing in A. Describe an efficient algorithm that finds the missing number. Try to describe an algorithm that uses only a constant amount of additional memory.

Proposition 1.1 (Gauss's sum). $\sum_{i=1}^{n} i = \frac{n(n-1)}{2}$

2. Sorted Array Fixed Point We are a given an array A of n integers that are sorted in increasing order. The range of the integers is not bounded in this case, in particular, numbers can be negative. Describe an efficient algorithm which finds an index i such that $A[i] = i$.

Bonus. Does your approach need to change if A contains some duplicate entries?

Consider the array $B = \left[A[i] - i\right]_{i=1}^{n}$. What are we looking for in this array?

Bonus. Notice that standard binary search fails on $A = [1, 1, 4, 6, 6]$. Then $B = [0, -1, 1, 2, 1]$.
Proposition 1.2. *For an index j, let $m = A[j]$. If $j < m$, then $i \notin (m, j]$.*

3. Egg testing The Empire State Building is a 102-story skyscraper, and we know that if we throw an egg from the K-th floor or higher, it will break. Unless the egg breaks, it can be collected and reused. We want to determine K, but use as few attempts (throws) as possible. What's the best strategy (minimizing the number of throws) if we have

a) one egg,

b) unlimited eggs,

c) two eggs,

d) **Bonus.** three eggs, or in general $e \in \mathbb{N}$ eggs?

(The eggs can be special, so nothing can be *a priori* assumed about K.)

Binary search is efficient because it rules out large chunks of the array at every step. But this can't work with only two eggs…right?

Suppose that we throw the first egg from floors m, $2m$, $3m$ etc. How many throws do we use in the worst case?

4. Laser There is a row of N buildings with $h_1, \ldots, h_n$ floors, and we need to demolish all of them. To that end, you found at home a demolition laser which is capable of firing *vertically* to destroy an arbitrary building or *horizontally* to destroy a given floor in all buildings (i.e., if you choose to destroy floor L, then the number of floors decreases by 1 for all buildings whose number of floors is $\geq L$). Develop an algorithm to determine minimum number of firings necessary to eliminate all buildings? (Beware that the maximum number of floors can be much more than N.)

Assume that every vertical firing targets the tallest remaining building and that every horizontal firing targets the ground floor. Clearly, there exists an optimal firing strategy which follows these rules. Now, let u and v represent the number of horizontal and vertical firings respectively. Define $f(v)$ to be the number of floors in the v^{th} tallest building. If h is sorted, then $f(v) = h_{n-v}$.

Proposition 1.3. *An optimal firing strategy will minimize $f(v) + v$.*

5. Submatrix search We are given an integer matrix A with n columns and m rows. Describe an efficient algorithm that finds a maximum submatrix of A consisting only of values 0 (i.e. the submatrix with largest area). Can you achieve complexity $O(nm)$?

Each row j has at most n intervals (i_1, i_2) of zero.

6. Fibonacci sequence Fibonacci sequence is defined using $F_0 = 0$, $F_1 = 1$, $F_n = F_{n-1} + F_{n-2}$.

a) What is the value of F_n for a given n?

b) Describe an algorithm for computing F_n using only $\Theta(\log_2 n)$ arithmetic operations.

a) Suppose that F_n takes the form r^n. What does r need to be? Does this work for $n = 0$ or $n = 1$?

b) Try representing the recursion in the form of matrix multiplication.