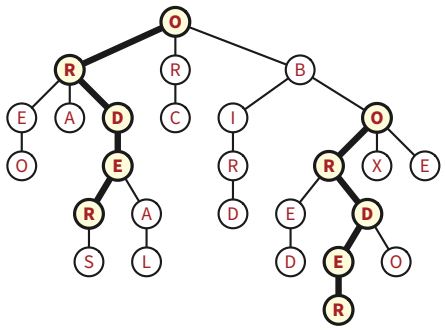
1. Prove that every integer (positive, negative, or zero) can be written in the form $\sum_i (-2)^i$, where the exponents i are distinct non-negative integers. For example:

$$42 = (-2)^6 + (-2)^5 + (-2)^4 + (-2)^0$$
$$25 = (-2)^6 + (-2)^5 + (-2)^3 + (-2)^0$$
$$17 = (-2)^4 + (-2)^0$$

2. Suppose you are given a rooted tree T, where each vertex is labeled with a symbol from some fixed alphabet. A ***downward path*** in T is a path from any node v to any descendant of v. The *label* of a downward path is the string formed by concatenating the labels of the vertices in order along the path.

   Describe an algorithm that finds the length of the longest string that is the label of *at least two different* downward paths in T.

   For example, given the following tree T as input, your algorithm should return the integer 5, which is the length of the string `ORDER`.



3. (a) ***Shifting*** a bit-string w removes some number of bits from one end of w and adds the same number of `0` bits at the other end. For example, shifting `10100101` to the right by 3 yields the bitstring `00010100`, and shifting `10100101` left by 2 yields `10010100`.

   Suppose we are given two bit-strings $A[1..n]$ and $B[1..n]$. Describe an algorithm that finds a shift A' of A that maximizes the number of indices i where $A'[i] = B[i] = 1$. For full credit your algorithm should run in $O(n \log n)$ time.

   For example, if $A =$ `001110011` and $B =$ `101000101`, your algorithm should return the integer 3, which is obtained by shifting A left by 2.

   (b) ***Rotating*** a bit-string w moves some number of bits from one end of w to the other. For example, rotating `10100101` to the right by 3 yields the bitstring `10110100`, and rotating `10100101` left by 2 yields `10010110`.

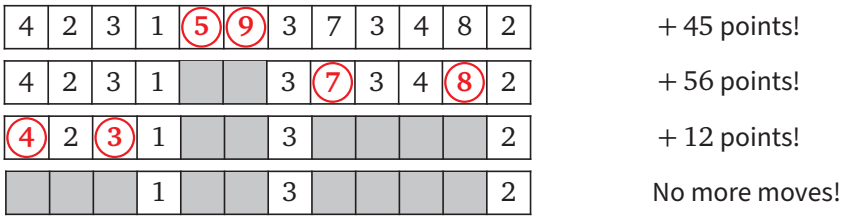
   Suppose we are given two bit-strings $A[1..n]$ and $B[1..n]$. Describe an algorithm that finds a *rotation* A' of A that maximizes the number of indices i where $A'[i] = B[i] = 1$. For full credit your algorithm should run in $O(n \log n)$ time.

Question 4 is on the other side of this page.

4. After being bombarded for months by TikTok ads, you finally break down and download the latest viral mobile puzzle game, Number Blast.

   Each Number Blast puzzle starts with a long row of numbered squares. On each turn, you select two squares *with no gaps between them*. After you make your choices, you earn points equal to the *product* of your two chosen numbers, and then your chosen squares *and all squares between them* are removed. Once a square is removed, it cannot participate in any future moves. The stage ends when no more moves are possible. Your goal is to earn as many points as possible.

   For example, the following sequence of turns earns a total of 113 points. (This is not necessarily the highest possible score for this sequence of numbers.) Gray squares indicate gaps; those squares were removed in earlier turns. At the end, there are no more legal moves, because all remaining numbers are separated by gaps.



   Describe and analyze an algorithm to find the maximum number of points you can earn from a given Number Blast puzzle. The input to your algorithm is an array $A[1..n]$ of positive integers.