KASINO

Observe that shuffling N cards in P compartments yields a concatenation of P unimodal sequences of numbers i.e. one big sequence in which numbers increase then decrease P times.

The algorithm proceeds by guessing the values of cards 1 through N in order, determining whether it's currently in the increasing or decreasing part of the sequence, based on the values of the previous two cards. If the values are currently increasing, we guess that the next card is the lowest valued card that hasn't yet been used (similarly for a decreasing sequence).

Imagine that the entire deck of cards is partitioned into 2P subsequences, one increasing, one decreasing etc. Analyzing the described algorithm, we can assert that the probability of correctly guessing any card in the first of the 2P subsequences is 1/(2P). The same probability is 1/(2P-1) for any card in the second subsequence etc. and 1 in the last subsequence.

Totaling over the 2P subsequences gives the following expected winnings for various values of P:

- $3.50 \cdot N$, when P equals 1
- $2.12 \cdot N$, when P equals 2
- $1.45 \cdot N$, when P equals 3
- $1.04 \cdot N$, when P equals 4
- $0.76 \cdot N$, when P equals 5

POLJA

We first compress the given coordinate system to a new, composed of $2N \times 2N$ "unit" squares. Observe that the geometry of rectangles is preserved i.e. what was previously a rectangle is also a rectangle in the new coordinate system. This allows us to use a $2N \times 2N$ matrix in which each unit square is colored with the color of the rectangle it's contained in.

Imagine using a horizontal line and moving it downwards, row by row, starting from the top row of the matrix. In each iteration, we can construct a histogram of squares resting on the line. By finding the largest rectangle of one color in each such histogram, we will have found the largest rectangle overall, since it must rest on some horizontal line.

One algorithm that finds the largest rectangle in a single histogram processes columns in left-to-right order and maintains a stack of potential solutions: a potential solution is a rectangle described by its height and leftmost column. Our goal is to extend all potential solutions as much to the right as possible. For each column we:

- if the stack is not empty and the color of the current column is different than the color of the potential solutions on the stack, empty the stack since we surely cannot extend it any farther to the right
- if the stack is not empty and the color of the current column matches that of the potential solutions on the stack, remove potential solutions off the top of stack while the current column is lower than the potential solution on top of the stack
- open a new potential solution

The time complexity of compressing the coordinate system is $\Theta(N^2)$, after which we perform N iterations, each solving a single histogram containing N columns in $\Theta(N)$ time. The overall worst-case (and average) time complexity of the solution is thus $\Theta(N^2)$, as well as the memory required.