### Kick Start 2020 - Round G

# Analysis: Merge Cards

## Test Set 1

In the i-th round, Panko will have **N** - i choices. In total, there are (**N** - 1)! possibilities. Using an exhaustive approach, the expected value can be computed by definition.

## Test Set 2

After each round, the number on each card equals to the sum of a subarray of **A**. In the last round, there are **N** - 1 cases:

- $\mathbf{A_1}$, $\mathbf{A_2} + \mathbf{A_3} + \ldots + \mathbf{A_N}$
- $\mathbf{A_1} + \mathbf{A_2}$, $\mathbf{A_3} + \mathbf{A_4} + \ldots + \mathbf{A_N}$

  ⋮
- $\mathbf{A_1} + \mathbf{A_2} + \ldots + \mathbf{A_i}$, $\mathbf{A_{i+1}} + \mathbf{A_{i+2}} + \ldots + \mathbf{A_N}$

  ⋮
- $\mathbf{A_1} + \mathbf{A_2} + \ldots + \mathbf{A_{N-1}}$, $\mathbf{A_N}$

The last round contributes a fixed number, $\mathbf{A_1} + \mathbf{A_2} + \ldots + \mathbf{A_N}$, to the answer. But in each case, the part contributed by the previous rounds is actually the sum of the answers of two variants of the original problem:

- Replace **A** by $\mathbf{A_1}, \mathbf{A_2}, \ldots, \mathbf{A_i}$
- Replace **A** by $\mathbf{A_{i+1}}, \mathbf{A_2}, \ldots, \mathbf{A_N}$

This part is then a weighted average of those **N** - 1 sums. However, each case is equally likely to occur. To reach a specific case in the last round, Panko has to avoid exactly one choice in each of the previous rounds. Thus arithmetic mean can be used here.

There are $O(\mathbf{N}^2)$ different subproblems. The answer of each of subproblem can be computed in $O(\mathbf{N})$ by [dynamic programming](). The overall time complexity is then $O(\mathbf{N}^3)$.

## Test Set 3

The above solution can be optimized to achieve $O(\mathbf{N}^2)$ time complexity by maintaining the prefix sums and suffix sums of the answers to the subproblems. But there is a faster approach that the $O(\mathbf{N}^2)$ part is in precomputation instead of having $O(\mathbf{N}^2)$ per test.

Let $\text{solve}_\mathbf{N}(\mathbf{A_1}, \mathbf{A_2}, \ldots, \mathbf{A_N})$ be a function that takes the **N** numbers written on the cards as parameters and returns the expected total score. It can be expressed as the average of **N** - 1 numbers. Each corresponds to a different choice in the first round. In particular, they are:

- $\mathbf{A_1} + \mathbf{A_2} + \text{solve}_{\mathbf{N}-1}(\mathbf{A_1} + \mathbf{A_2}, \mathbf{A_3}, \ldots, \mathbf{A_i}, \mathbf{A_{i+1}}, \ldots, \mathbf{A_{N-1}}, \mathbf{A_N})$
- $\mathbf{A_2} + \mathbf{A_3} + \text{solve}_{\mathbf{N}-1}(\mathbf{A_1}, \mathbf{A_2} + \mathbf{A_3}, \ldots, \mathbf{A_i}, \mathbf{A_{i+1}}, \ldots, \mathbf{A_{N-1}}, \mathbf{A_N})$

  ⋮
- $\mathbf{A_i} + \mathbf{A_{i+1}} + \text{solve}_{\mathbf{N}-1}(\mathbf{A_1}, \mathbf{A_2}, \mathbf{A_3}, \ldots, \mathbf{A_i} + \mathbf{A_{i+1}}, \ldots, \mathbf{A_{N-1}}, \mathbf{A_N})$

  ⋮
- $\mathbf{A_{N-1}} + \mathbf{A_N} + \text{solve}_{\mathbf{N}-1}(\mathbf{A_1}, \mathbf{A_2}, \mathbf{A_3}, \ldots, \mathbf{A_i}, \mathbf{A_{i+1}}, \ldots, \mathbf{A_{N-1}} + \mathbf{A_N})$

By using mathematical induction with $\text{solve}_2(\mathbf{A_1}, \mathbf{A_2}) = \mathbf{A_1} + \mathbf{A_2}$ as the base case, it can be shown that $\text{solve}_\mathbf{N}(\mathbf{A_1}, \mathbf{A_2}, ..., \mathbf{A_N})$ is a linear combination of $\mathbf{A_1}, \mathbf{A_2}, ..., \mathbf{A_N}$, which means $\text{solve}_\mathbf{N}(\mathbf{A_1}, \mathbf{A_2}, ..., \mathbf{A_N}) = k_{\mathbf{N}, 1} \times \mathbf{A_1} + k_{\mathbf{N}, 2} \times \mathbf{A_2} + ... + k_{\mathbf{N}, \mathbf{N}} \times \mathbf{A_N}$ where $k_{i, j}$ are constants. If all $k_{i, j}$ are precomputed, the answer for each test can be computed in time complexity $O(\mathbf{N})$.

Starting from $k_{2, 1} = k_{2, 2} = 1$, $k_{i, j}$ can be computed in increasing order of i. The transition formulas can be obtained by transforming the formula of $\text{solve}_\mathbf{N}(\mathbf{A_1}, \mathbf{A_2}, ..., \mathbf{A_N})$. For example, the transition formula of $k_{\mathbf{N}, 1}$ can be obtained by replacing $\text{solve}_{\mathbf{N} - 1}(...)$ and $\text{solve}_\mathbf{N}(...)$ by the expressions with $k_{i, j}$, then comparing the coefficients of the $\mathbf{A_1}$ term.

The j-th card will become either (part of) the (j - 1)-th card or (part of) the j-th card after a round. Correspondingly, only the coefficients of the $k_{i - 1, j - 1}$ term, the $k_{i - 1, j}$ term and the constant term can be non-zero in the transition formula of $k_{i, j}$. By grouping the like terms, the number of terms in each transition formula is reduced to at most 3. Then the overall time complexity becomes $O(\mathbf{N}^2)$.

The error analysis is straightforward since $\mathbf{A_i}$ cannot be negative. It should not be a problem in most of the implementations.