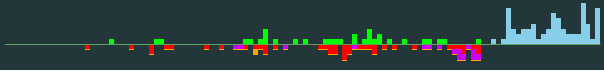
L: Lock In

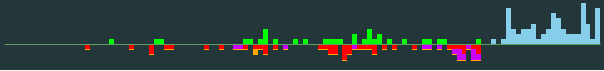
Problem author: Mike de Vries



Problem: Win the ranking game at least 1% of the time.

L: Lock In

Problem author: Mike de Vries

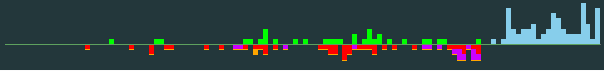


Problem: Win the ranking game at least 1% of the time.

Observation: For a game with n numbers to be placed, we may always assume that we must place n random numbers chosen from $1, \ldots, m$.

L: Lock In

Problem author: Mike de Vries



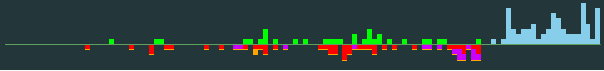
Problem: Win the ranking game at least 1% of the time.

Observation: For a game with n numbers to be placed, we may always assume that we must place n random numbers chosen from $1, \ldots, m$.

Solution: We can determine the optimal solution recursively.

L: Lock In

Problem author: Mike de Vries



Problem: Win the ranking game at least 1% of the time.

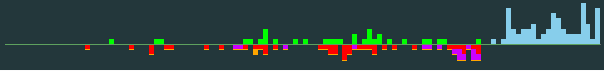
Observation: For a game with n numbers to be placed, we may always assume that we must place n random numbers chosen from $1, \ldots, m$.

Solution: We can determine the optimal solution recursively.

If we need to place a single number, the game is always winnable.

L: Lock In

Problem author: Mike de Vries



Problem: Win the ranking game at least 1% of the time.

Observation: For a game with n numbers to be placed, we may always assume that we must place n random numbers chosen from $1, \ldots, m$.

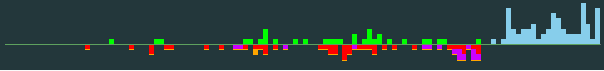
Solution: We can determine the optimal solution recursively.

If we need to place a single number, the game is always winnable.

For each of these numbers, we must compute the optimal placement, placing k at index j is given the product by the winning chance of games of size $j-1$, and $n-j$, and the chance that $j-1$ of the rest of the randomly chosen numbers are at most $j-1$.

L: Lock In

Problem author: Mike de Vries



Problem: Win the ranking game at least 1% of the time.

Observation: For a game with n numbers to be placed, we may always assume that we must place n random numbers chosen from $1, \ldots, m$.

Solution: We can determine the optimal solution recursively.

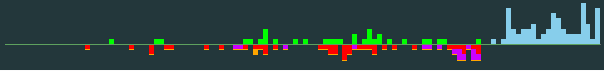
If we need to place a single number, the game is always winnable.

For each of these numbers, we must compute the optimal placement, placing k at index j is given the product by the winning chance of games of size $j-1$, and $n-j$, and the chance that $j-1$ of the rest of the randomly chosen numbers are at most $j-1$.

Pre-compute the optimal strategy and follow it for each game.

L: Lock In

Problem author: Mike de Vries



Problem: Win the ranking game at least 1% of the time.

Observation: For a game with n numbers to be placed, we may always assume that we must place n random numbers chosen from $1, \ldots, m$.

Solution: We can determine the optimal solution recursively.

If we need to place a single number, the game is always winnable.

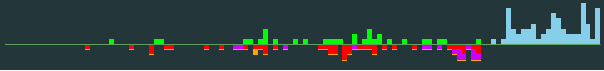
For each of these numbers, we must compute the optimal placement, placing k at index j is given the product by the winning chance of games of size $j-1$, and $n-j$, and the chance that $j-1$ of the rest of the randomly chosen numbers are at most $j-1$.

Pre-compute the optimal strategy and follow it for each game.

The resulting probability of winning is $\frac{374337547685188753}{15703912738882368000} \approx 0.023837$. The resulting probability of winning less than 100 games out of 10 000 is less than 1 in 10^{24}.

L: Lock In

Problem author: Mike de Vries



Problem: Win the ranking game at least 1% of the time.

Observation: For a game with n numbers to be placed, we may always assume that we must place n random numbers chosen from $1, \ldots, m$.

Solution: We can determine the optimal solution recursively.

If we need to place a single number, the game is always winnable.

For each of these numbers, we must compute the optimal placement, placing k at index j is given the product by the winning chance of games of size $j-1$, and $n-j$, and the chance that $j-1$ of the rest of the randomly chosen numbers are at most $j-1$.

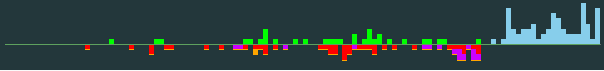
Pre-compute the optimal strategy and follow it for each game.

The resulting probability of winning is $\frac{374337547685188753}{15703912738882368000} \approx 0.023837$. The resulting probability of winning less than 100 games out of 10 000 is less than 1 in 10^{24}.

Solution: Optimal strategy is not necessary. For example, you can also linearly interpolate to make a good enough guess. There are many accepted solutions!

L: Lock In

Problem author: Mike de Vries



Problem: Win the ranking game at least 1% of the time.

Observation: For a game with n numbers to be placed, we may always assume that we must place n random numbers chosen from $1, \ldots, m$.

Solution: We can determine the optimal solution recursively.

If we need to place a single number, the game is always winnable.

For each of these numbers, we must compute the optimal placement, placing k at index j is given the product by the winning chance of games of size $j-1$, and $n-j$, and the chance that $j-1$ of the rest of the randomly chosen numbers are at most $j-1$.

Pre-compute the optimal strategy and follow it for each game.

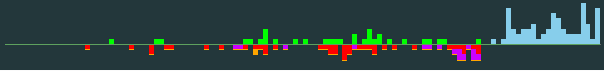
The resulting probability of winning is $\frac{374337547685188753}{15703912738882368000} \approx 0.023837$. The resulting probability of winning less than 100 games out of 10 000 is less than 1 in 10^{24}.

Solution: Optimal strategy is not necessary. For example, you can also linearly interpolate to make a good enough guess. There are many accepted solutions!

Fun fact: With 10 uniformly random real numbers in $[0, 1]$, the optimal probability of winning is about 0.020403.

L: Lock In

Problem author: Mike de Vries



Problem: Win the ranking game at least 1% of the time.

Observation: For a game with n numbers to be placed, we may always assume that we must place n random numbers chosen from $1, \ldots, m$.

Solution: We can determine the optimal solution recursively.

If we need to place a single number, the game is always winnable.

For each of these numbers, we must compute the optimal placement, placing k at index j is given the product by the winning chance of games of size $j-1$, and $n-j$, and the chance that $j-1$ of the rest of the randomly chosen numbers are at most $j-1$.

Pre-compute the optimal strategy and follow it for each game.

The resulting probability of winning is $\frac{374337547685188753}{15703912738882368000} \approx 0.023837$. The resulting probability of winning less than 100 games out of 10 000 is less than 1 in 10^{24}.

Solution: Optimal strategy is not necessary. For example, you can also linearly interpolate to make a good enough guess. There are many accepted solutions!

Fun fact: With 10 uniformly random real numbers in $[0, 1]$, the optimal probability of winning is about 0.020403.

Statistics: 160 submissions, 32 accepted, 70 unknown