

Sure Bet

Tomaž Hočevar

The problem is asking us to maximize the value $\min(\sum a_i - n_a - n_b, \sum b_j - n_a - n_b)$, where a_i represent the odds for placed bets on the first outcome and n_a the number of them, while b_j and n_b stand for the corresponding values for the second outcome.

Brute force. The straight-forward way to solve the problem is to simply try all possible subsets of bets. There is 2^{2n} of them, which is small enough for $n \leq 10$ and solves the first subproblem.

Greedy. Let's simplify the formula we're trying to maximize. If we subtract 1 from every quota, the formula simplifies to $\min(\sum a_i - n_b, \sum b_j - n_a)$. Note that the two outcomes are independent for a fixed n_a and n_b — the exact set of odds we choose for the first outcome does not affect the optimal choice of bets for the second outcome. Therefore, it makes sense to sort the values a_i and b_i . We can choose the n_a largest a_i and n_b largest values b_i for every pair of n_a and n_b . This gives us a $O(n^2)$ solution for the second subproblem.

Ternary search. Consider what is the score for a fixed value of n_a and different values of n_b . As we increase n_b the score grows until the term $\sum b_j - n_a$ becomes greater than $\sum a_i - n_b$ at which point it starts to decrease. Therefore, we can use ternary search to find its maximum or a binary search over the list of differences in score for consecutive values of n_b . Precompute prefix sums to compute the score for a given pair n_a and n_b in $O(1)$. Time complexity of this solution is $O(n \log n)$.

Linear solution. There is in fact a linear solution assuming the values are already sorted. We can try to visualize what is going on as we try different values of n_a and n_b . For every a_i that we bet on, we increase the first value in the minimum by a_i and decrease the other one by 1, while trying to maintain the lower one as large as possible. This would suggest that if n_b is the optimal choice for n_a then as we increase n_a to $n_a + 1$ we should also increase n_b or keep it at the same value but not decrease it.

Let $A = \sum_{n_a} a_i - n_b$ and $B = \sum_{n_b} b_j - n_a$ where n_b is the optimal choice for n_a . As we increase n_a by 1, we get $A' = A + a$ and $B' = B - 1$. We will handle two cases:

1. $A' \geq B'$. Because B' is already the smaller of the two values decreasing n_b would only make the minimum smaller.
2. $A' < B'$. Suppose that decreasing n_b would give us a better solution. The new score in this case $\min(A' + 1, B' - b)$ would have to be larger than the previous one $\min(A', B') = A'$. This would imply $A' < B' - b$. Rewriting it we get $A + 1 < B - b - a$. This means that n_b is not the optimal solution for n_a as we assumed because we could decrease n_b and obtain a better solution for n_a ; we have a contradiction.

We don't have to restart the search for n_b from scratch for every value of n_a in increasing order. Instead, we can simply increase n_b until we find the maximum for a given n_a . For $n_a + 1$ we start the search for n_b from where we finished previously, overall making a single pass over all values of n_b .