Problem Tutorial: "Joy"

If our position in the queue is fixed, we can use DP. Consider a binary tree with n leaves, and let $f(i, j)$ be the probability that person (leaf) j wins in the subtree of vertex i. Time complexity of such DP is $O(n^2)$. However, if we try all n positions independently, we'll arrive at an $O(n^3)$ solution, which is too slow.

Note that if we fix our position, we know that to become the champion, we have to beat the winners of some subtrees. If we know DP values for these subtrees, we can calculate the probability of becoming the champion in $O(n)$, totalling in $O(n^2)$ for n starting positions.

Finally, note that there are not so many different subtrees we might want to beat. For example, if $n = 16$, we might want to beat all segments of 1 and 2 people, only the following segments of 4 people: $a_{1..4}, a_{5..8}, a_{9..12}, a_{4..7}, a_{8..11}, a_{12..15}$, and the following segments of 8 people: $a_{1..8}, a_{8..15}$. The number of interesting segments is only twice the number of segments in the original DP for one tree, and time complexity of calculating DP for interesting segments is still $O(n^2)$.