Parentrises - Editorial

Bogdan Ciobanu, Andrei-Costin Constantinescu

11 July 2018

1 Solving P=1

1.1 $O(N^3)$

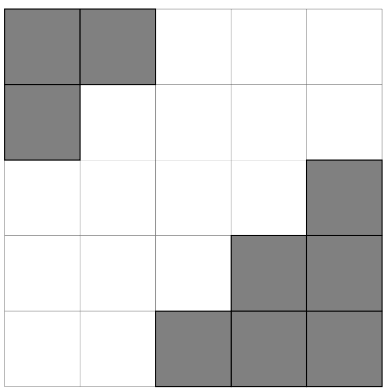
To check whether a string admits a coloring or not we can derive a simple dynamic programming solution, as follows: D_{i,s_1,s_2} is true if we can color the first i characters of S such that the balance of the Red and Green resulting string is s_1 and the balance of the Blue and Green resulting string is s_2. Its recurrence is pretty straight forward, if we can obtain (i, s_1, s_2) and the $(i+1)^{th}$ character is $'('$, then the following states are also valid: $(i+1, s_1+1, s_2)$ (in case we color it in Red), $(i+1, s_1, s_2+1)$ (in case we color it in Blue) and $(i+1, s_1+1, s_2+1)$ (in case we color it in Green). In the other case, the results are the same, except the sign will change for the balance quota. There is a solution if and only if $(N, 0, 0)$ is a valid state. In that case, we can use the table to reconstruct the answer backwards.

1.2 $O(\frac{N^3}{64})$

We can improve the previous solution by means of a bitset.

1.3 $O(N)$

Plotting the pairs (s_1, s_2) in a lattice for all D_i, we can prove that they'll form a convex set. More specifically, at any point in time, the set of valid (s_1, s_2) is a square cut above some diagonal d_1 and below some other d_2.



If we label the diagonals from 0 to $2N-2$, then we can determine the size of the square $L = \frac{d_2}{2} - 2$ (we consider d_2 as the right-most diagonal of the square). We also consider $d_1 = -1$ or $d_2 = 2N-1$ in which case we ignore it.

We can inductively prove that this holds. The base case holds, as D_0 is a square of length 1 with nothing cut. If the hypothesis holds for D_k, then we can prove it also holds for D_{k+1}. If $S_{k+1} =$ '(' , then $d_1 := d'_1 + 1$ and $d_2 := d'_2 + 2$. In the other case, $d_1 := d'_1 - 2$ and $d_2 := d'_2 - 1$. If at any point in time d_1 becomes less than -1, then we label it back to -1. For a solution to exist, d_2 must be positive at all times and d_1 must be different than -1 at the end.

2 Solving P=2

We can derive an $O(N^3)$ dynamic programming recurrence to count the number of colourable strings of length N, following up from the observations from the $O(N)$ solution for the other case. The state consists of the length of the string and the values of d_1 and d_2.