Problem Tutorial: "Kilk"

WLOG assume $x \le y$. It's not hard to compute the smallest possible length of the longest substring consisting of equal letters: in fact, it is $k = l(x, y) = \left\lfloor \frac{x+y}{x+1} \right\rfloor$.

Let's fix some value of k and find the required number of strings for all pairs (x, y) such that $l(x, y) = k$. Let $a_k(x, y)$ be the number of strings that have x letters '`a`' and y letters '`b`', don't have substrings of equal letters of length more than k, and end with '`a`'. Let $b_k(x, y)$ be defined similarly, except that here we count strings ending with '`b`'. Then, the answer for a pair (x, y) is $a_{f(x,y)}(x, y) + b_{f(x,y)}(x, y)$.

Trying all possible lengths of the substring of equal letters at the end of the string, we get the following formulas:

- $a_k(x, y) = \sum\limits_{i=1}^{\min(k,x)} b_k(x - i, y)$;
- $b_k(x, y) = \sum\limits_{i=1}^{\min(k,y)} a_k(x, y - i)$.

Computed in a straightforward way, we can compute the answers for all pairs (x, y) with $x, y \le n$ in $O(n^4)$.

We can use the following two optimizations to speed it up to $O(n^2 \log n)$:

- use prefix sums to find $a_k(x, y)$ faster, or just notice that $a_k(x, y) = a_k(x - 1, y) + b_k(x - 1, y) - b_k(x - 1 - k, y)$ (almost follows from the definition);
- note that we only need $x \le \frac{2n}{k}$, therefore, the number of interesting DP states is about $\sum\limits_{k=1}^{n} \frac{2n^2}{k} = O(n^2 \log n)$.

Time limit is a bit (unnecessarily) strict, so one needs to be careful with implementation.