Let $G = (V_1 \cup V_2, E)$ be a bipartite graph, where $|V_1| = |V_2| = n$ and $|E| = m$. Let the vertices of V_1 be denoted by $v_1, \ldots, v_n$ and let the vertices of V_2 be denoted by $w_1, \ldots, w_n$. A matching M is *planar* if every pair of edges in M do not cross. More formally, all pair of edges (v_i, w_j) and $(v_{i'}, w_{j'})$ in M satisfy the requirement that $i < i'$ if and only if $j < j'$.

Describe an $O(m^{5/2})$ algorithm that returns the minimum number of planar matchings $M_1, \ldots, M_k$ such that each edge in E lies in *exactly* one matching M_i.

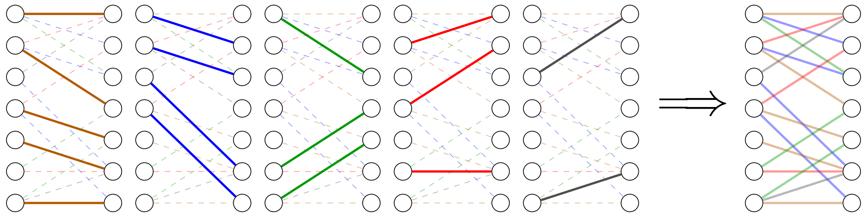

Figure 1: *A collection of five planar matchings.*

There are at least four approaches.

Approach 1. *Reduce to a problem that you have previously seen, perhaps in the tutorials... this should have running time $O(m^{5/2})$.*

Approach 2. *Use Dilworth's theorem. Construct an appropriate poset, and use the chain-antichain duality. The minimum number of **chains** that cover the poset should correspond to the maximum length **antichain**. This should have running time $O(m^2)$.*

Approach 3. *Use dynamic programming and reduce to the longest decreasing subsequence problem. This should have running time $O(m \log m)$.*

Approach 4. *Use ordered dictionaries or van Emde Boas trees for the competitive programming enthusiasts. This should have running time $O(m \log \log n)$.*

Solution. Solution goes here...