

Analysis: Intranets

We use terminology for [graphs](#). The vertices are the M machines and the edges are the $\binom{M}{2}$ links.

Test Set 1

Let's consider the process of assigning priorities to the edges one by one, from the highest priority to the lowest. For simplicity, we number the i -th highest priority as priority i so that the smaller the number, the higher the priority.

Suppose we have assigned the highest i priorities to i edges, and we need to assign the priority $(i + 1)$ to a new edge. How can we do?

We have three types of choices. Suppose the new edge with the priority $(i + 1)$ is (u, v) . Let S_i be the set of vertices that the edges with highest i priorities connect with.

1. If the vertices u and v don't occur in the previously assigned edges. I.e. $u \notin S_i, v \notin S_i$, the number of such edges is $\binom{M-|S_i|}{2}$. In this case, u and v form a new intranet. So, the number of intranets increase by 1, and $|S_{i+1}|$ equals $|S_i| + 2$.
2. If $u \in S_i, v \notin S_i$, the number of such edges is $|S_i| \cdot (M - |S_i|)$. In this case, v joins the intranet in which u is located. So, the number of intranets does not change, and $|S_{i+1}|$ equals $|S_i| + 1$.
3. If $u \in S_i, v \in S_i$, the number of such edges is $\binom{|S_i|}{2} - i$. In this case, the edge is not activated. So, the number of intranets do not change, and $|S_{i+1}|$ equals $|S_i|$.

Then, we can use a dynamic programming approach to solve Test Set 1. Let $\text{dp}(i, j, k)$ denote the probability that we assign the highest i priorities to the edges, the set of vertices introduced by these i edges has the size of j , and k intranets have been formed. The transition function can be deduced from the discussion above. The time complexity is $O(M^4)$ and it is enough to pass Test Set 1.

We can speed this up to $O(M^2)$ by dropping i from the keys. Let $\text{dp}(j, k)$ denote the probability that after assigning the highest i priorities for some i , the set of vertices introduced by these i edges has the size of j , and k intranets have been formed. The transition is to assign the highest priority **among the edges of the types 1 and 2 above**. The probability of there being a new intranet is

$$\frac{\binom{M-|S_i|}{2}}{\binom{M-|S_i|}{2} + |S_i| \cdot (M - |S_i|)} \text{ as we are no longer interested in priorities of the edges of the type 3 above.}$$

Observing the graph

From the solution for Test Set 1, we see that each intranet corresponds to a pair of vertices (u, v) such that both u and v activate the edge (u, v) . This fact can also be proved directly, and we describe here.

Let's fix an assignment of priorities and consider the directed graph where the vertices are the machines and the edges are (u, v) such that machine u uses the links connecting machines u and v . Since this graph is a functional graph (each vertex has outdegree 1), each connected component contains exactly one cycle (and possibly some chains of nodes leading into the cycle).

A crucial observation is that the length of a cycle cannot be 3 or more. Assume the contrary: that vertices u_1, \dots, u_c ($c \geq 3$) form a cycle in this order, and let the priority of the link connecting u_i and u_{i+1} be p_i (indices are modulo c). Note that those links are distinct. Since machine u_i uses the link with highest priority, we have $p_i < p_{i-1}$. This gives us $p_1 > p_2 > \dots > p_c > p_1$, a contradiction. As a self loop is also impossible, we conclude that every cycle in the graph has length 2.

Test Set 2

Let's call the set of edges activated by both of their endpoints an *active matching* as they form a [matching](#). Now the problem is to compute the probability that the size of the active matching is exactly \mathbf{K} .

For a matching X , let $f(X)$ be the probability that the active matching is X . Since it is not easy to compute $f(X)$ directly, we apply [inclusion-exclusion principle](#) technique and so let $g(X)$ be the probability that the active matching **contains** X , that is, $g(X) = \sum_{Y \supseteq X} f(Y)$. By inverting this, we have $f(X) = \sum_{Y \supseteq X} (-1)^{|Y|-|X|} g(Y)$. Our answer is then calculated as follows:

$$\sum_{|X|=\mathbf{K}} f(X) = \sum_{|X|=\mathbf{K}} \sum_{Y \supseteq X} (-1)^{|Y|-|X|} g(Y) = \sum_{|Y| \geq \mathbf{K}} \binom{|Y|}{\mathbf{K}} (-1)^{|Y|-\mathbf{K}} g(Y) = \sum_{i \geq \mathbf{K}} \binom{i}{\mathbf{K}} (-1)^{i-\mathbf{K}} \sum_{|X|=i} g(X)$$

All that remains is to compute $g(X)$. Of course, this depends only on $|X|$, and we want to compute it when $|X| = i$, for each $i = \mathbf{K}, \mathbf{K} + 1, \dots, \mathbf{M}$. There are $i!$ possible orders of priorities assigned to X . Let's fix one of them and let the edges in X be $(u_1, v_1), (u_2, v_2), \dots, (u_i, v_i)$ from the lowest priority to the highest. Then the condition that the active matching contains X is equivalent to, for each $j = 1, 2, \dots, i$, edge (u_j, v_j) has the highest priority among edges touching at least one of $u_1, v_1, \dots, u_j, v_j$. Hence we have

$$g(X) = i! \prod_{j=1}^i \frac{1}{\binom{\mathbf{M}}{2} - \binom{\mathbf{M}-2j}{2}}.$$

Here we note that the denominators can be factorized to see that they are not divisible by $10^9 + 7$. The number matchings of size i is $\frac{1}{i!2^i} \frac{\mathbf{M}!}{(\mathbf{M}-2i)!}$, and this completes the $O(\mathbf{M})$ time solution (the divisions can be done efficient by using the factorization, though this is not necessary).

In fact, we can find the answers for $K = 1, \dots, \lfloor \frac{\mathbf{M}}{2} \rfloor$ at the same time in $O(\mathbf{M} \log \mathbf{M})$ time: The transformation from g to f can be represented by a convolution and we can use FFT.