Code Jam to I/O for Women 2021 - Code Jam to I/O for Women

Analysis: Introductions Organization

Test Set 1

In the first Test Set, the number of people in a new team is so small that we can use brute force. Even in the worst case, we only have five people: three managers and two non-managers. Since an introduction ties up three people, at most one introduction can take place in each minute. Moreover, there are only $\binom{3}{1} \times \binom{4}{2} = 18$ possible groups of an introducing manager plus two other people, and only some of these will meet the conditions needed for an introduction: the introducing manager must know the other two people, and the other two people must not know each other (or else there would be no point in introducing them).

We can use an exhaustive search to explore all ways of performing an introduction each minute until our target people are acquainted, or until we run out of valid introductions to make, and then return the smallest number of introductions we saw in a valid answer (or -1 if there is no answer). The bounds on the number of possible introductions and on the size of the answer (it cannot exceed 3 in this test set) ensure that the space to explore is small, and so our solution will run very quickly.

Test Set 2

To solve Test Set 2 we can model the problem as a graph. Every person in the team is a node, and we connect two different nodes with an edge if the represented people know each other when the team was formed. The key insight to solve this test set is to realize that an introduction changes the state from graph G to a graph H that is G with one extra edge, and the extra edge directly connects two nodes that are connected in G by a path of length 2 with an intermediate node that is a manager. Let $D_G(v, w)$ be the distance in G between nodes v and w considering just paths that only use managers as intermediate nodes. In what follows, when we say path or distance, we refer to those terms with the restriction of only using managers as intermediate nodes.

Following the reasoning in the previous paragraph, $D_G(v,w) - 1 \geq D_H(v,w)$ — that is, after an introduction, the distance between two nodes decreases by at most 1. Moreover, if $D_G(v,w) > 1$ — that is, v and w do not know each other — and there is a path of managers between them, there is always a way to make an introduction such that the resulting H has $D_H(v,w) = D_G(v,w) - 1$. Let $[v = x_0, x_1, x_2, \dots, x_{D_G(v,w)-1}, x_{D_G(v,w)} = w]$ be a minimum path that between v and w in G. Since v and w do not know each other, x_1 is neither of them, and therefore an intermediate step in the path, so x_1 is a manager. If x_1 introduces $v = x_0$ and x_2, the path $[v = x_0, x_2, \dots, x_{D_G(v,w)-1}, x_{D_G(v,w)} = w]$ of length $D_G(v,w) - 1$ is formed in the resulting H, so $D_H(v,w) = D_G(v,w) - 1$.

The goal of the problem is to use introductions to reduce the distance between a given pair of nodes v and w to 1 (the minimum possible distance). So, if we could only do one introduction at a time, the answer would be exactly $D_{G_0}(v,w)$ in the initial graph G_0. Since time considerations don't change whether it is possible for v and w to meet, we can already say that v and w can meet if and only if there is a path connecting them in the initial graph.

Let us now consider simultaneous introductions. Let $G_0, G_1, \dots, G_k$ be a list of graphs where G_i models the state of the team at time i in an optimal way to make v and w meet (meaning k is the answer for this pair). G_0 is the original team and v and w are connected directly by an edge only in G_k. For convenience, let $D_i = D_{G_i}(v,w)$. We can bound D_{i+1} in terms of D_i as follows:

each edge (x, y) on G_{i+1} either exists in G_i or comes from an introduction that happens between minutes i and $i+1$, meaning there is a path of length 2 $[x, m, y]$ in G_i where m is a manager. Now fix i and consider the path $[v = x_0, x_1, \ldots, x_{D_{i+1}-1}, x_{D_{i+1}} = w]$ between v and w in G_{i+1}. Consecutive edges share a person, so at most one of each pair can come from an introduction and the other one must exist in G_i. This gives us a bound $D_i \le D_{i+1} + \lceil (D_{i+1}/2) \rceil$. Let us show that this bound is tight and always reachable: if we are at state G_i, and $D_i > 1$, we can take a minimum path $[v = y_0, y_1, \ldots, y_{D_i-1}, y_{D_i} = w]$ between v and w in G_i and have y_{3j+1} to introduce y_{3j} and y_{3j+2} for each j between 0 and $\lceil ((D_i - 2)/3) \rceil$. Those introductions at minute i yield $D_{i+1} = D_i - \lceil ((D_i - 2)/3) \rceil$ and make exactly $\lceil (D_{i+1}) \rceil$ edges in G_{i+1} come from an introduction.

Given all the theory above, the answer for a given pair v and w for which $D_{G_0}(v, w)$ is undefined (that is, they are not connected) is impossible, and is otherwise $f(D_{G_0}(v, w))$ where $f(x)$ is the number of steps it takes for $x := x - \lceil ((x - 1)/3) \rceil$ to reach $x = 1$. Notice that it takes a logarithmic number of steps to reach 1, so we can just calculate f on demand, or we can memoize all the needed values (for x between 1 and $\mathbf{N}$). Most shortest-path algorithms can be modified slightly to calculate the distance D_0 considering only managers as intermediate nodes as above, but the simplest must be [Floyd-Warshall](). Floyd-Warshall computes the shortest paths between every pair of nodes by considering each possible intermediate node in its outermost loop. So, simply restricting that loop to iterate over managers (up to $\mathbf{M}$ instead of $\mathbf{M}$+$\mathbf{N}$) is enough to calculate, in time $O(\mathbf{M} \times (\mathbf{N} + \mathbf{M})^2)$, the distance D_{G_0} for every pair of nodes. After running Floyd-Warshall and memoizing f once, we can get the answer for every pair in constant time, so the algorithm takes $O(\mathbf{M} \times (\mathbf{M} + \mathbf{N})^2 + \mathbf{P})$ time overall.