Code Jam to I/O for Women 2018 - Code Jam to I/O for Women

Analysis: CEO Search

CEO Search: Analysis

Small dataset

In the Small dataset, there are at most 10 existing employees, so one viable approach is as follows. We will start by supposing that the new CEO has the minimum experience level possible: one level higher than the highest level among all existing employees. Then, we can check all possible assignments of employees to potential managers of higher levels than those employees. (Throughout this analysis, for convenience, we will use "manager" to mean "direct manager"). If any of the assignments is valid — that is, nobody is managing too many employees — then we have solved the problem. Otherwise, we try again with a CEO of one level higher, and so on. This process is bounded, since the answer cannot possibly be larger than 11; the existing employees are all level 10 or lower, so a level 11 CEO could manage all of them personally.

At first, it may seem that the search space is too large. When we have one employee of each level between 1 and 10, for example, and we try to add a new CEO of level 11, there are 10! possible sets of assignments. (The level 1 employee has 10 potential managers, the level 2 employee has 9 potential managers, and so on.) But the cases like this one with large search spaces are also easy to solve; note that any assignment is valid in our example, since it is impossible to give a manager too many employees. On the other hand, when we have a case like five level 0s, four level 1s, and one level 2, and we try to add a new CEO of level 3, it turns out that no set of assignments is valid... but there are so few possible sets of assignments that it is easy to check them all and move on to trying a new CEO of level 4.

Large dataset

Suppose that we have chosen a new CEO of some level (which may or may not turn out to work), and we have started trying to assign managers in some way; not everyone necessarily has a manager yet. Suppose that an employee A of level L_A is managing an employee B of level L_B, and there is some other employee C of level L_C (not managed by A) such that $L_A > L_C > L_B$. Then we can safely have A manage C instead of B. If employee C previously had a manager D, that manager is of a high enough level to manage employee B, so we can have D manage B and we are no worse off than before. If employee C did not yet have a manager, then we are better off than before. The same logic holds if employee B stands for an empty space rather than an employee.

This observation suggests a top-down approach to the problem. We can work from the highest-level employees to the lowest-level ones, and apply a simple rule: each employee should manage as many other employees as they can, and those employees should be of the highest levels available among all remaining unmanaged employees.

We can simplify this approach by thinking in terms of levels instead of employees. We will start at the top level, which has the new CEO of level X; the new CEO creates X management "slots". Then, let us consider the next highest level, which has $\mathbf{N_L}$ employees of level $\mathbf{E_L}$. If $\mathbf{N_L}$ > X, then our plan does not work and we need a CEO with a higher level. Otherwise, we have used up $\mathbf{N_L}$ of our X slots, but the employees from this level have added $\mathbf{N_L} \times \mathbf{E_L}$ more slots. We can continue in this way, abandoning a plan whenever we do not have enough slots to manage all of the employees at a given level.

For the Large dataset, the CEO level may be as large as the largest possible $\mathbf{N_i}$ value plus 1, so we may not have time to check all of them. We can speed up the process by binary searching on the level of our new CEO, with an inclusive lower bound equal to $\mathbf{E_L}$ + 1, and an inclusive upper bound equal to the lower bound or the maximum $\mathbf{N_i}$ allowed by the problem, whichever is greater. (If there are 10^{12} level 0 employees, for example, the answer could be as large as 10^{12}.) If we try a possible value and find that it works, we can make that value our new upper bound; if a value does not work, we can set the lower bound to that value plus 1. Each search takes O(**L**) time, and the binary search adds another factor of log(max($\mathbf{N_i}$)), so the running time is O(**L** × log(max($\mathbf{N_i}$))).

This is fast enough to solve the Large dataset, but we can do even better by cutting out the binary search entirely! We can proceed through the levels as before, but without including the new CEO or trying to guess their level. Whenever we encounter employees with no possible managers, we add them to a count and then move on. (This is definitely the case for every employee of level $\mathbf{E_L}$, and it may be true of employees at other levels as well.) The final answer is then either that count (since all of those unmanaged employees must be personally managed by the new CEO), or $\mathbf{E_L}$ + 1, whichever is greater.

It is also possible to work from the bottom up instead, using the same idea, or to apply Hall's theorem to get a closed-form expression of the solution; we leave those as exercises. In any case, our running time is now O(**L**).