Kick Start 2017 - Round D

Analysis: Sightseeing

Go Sightseeing: Analysis

Small dataset

There are at most 15 cities in which you can go sightseeing, and in each of those cities, we can either go sightseeing or not, so there are only 2^{15} possibilities to consider. A brute force approach that tries each of those 2^{15} possibilities is fast enough. For each possibility, we can compute the earliest possible time of arrival in city **N** and then find the possibility with the greatest number of sightseeing trips that arrives at city **N** on time.

It can be observed that the time of arrival in any city i + 1 is only dependent on T_w, the time at which we start waiting for the bus in city i. This can be expressed as *Arrival*(i + 1, T_w) = T' + $\mathbf{D_i}$ - ((T' - $\mathbf{S_i}$) mod $\mathbf{F_i}$), where T' is max(T_w, $\mathbf{S_i}$). The arrival time at the final city can thus be computed by iteratively applying the *Arrival*() function to each city in succession. T_w for city i will equal *Arrival*(i) if we do not go sightseeing in city i, and *Arrival*(i) + $\mathbf{T_s}$ if we do.

Large dataset

A brute force approach will not work for the large dataset as there are now 2000 cities. However, as noted above, the time of arrival in any city is only dependent on the time that we start waiting at the previous city. This allows us to use dynamic programming to solve this problem.

One option that might come to mind is to compute m(i, j), the maximum number of cities that you can go sightseeing in if you have reached city i by time j. This can easily be expressed as a recurrence relation on smaller values of i and j. However, since the times can be as large as 10^9, this approach is not possible. Instead, we should try to compute f(i, j), the earliest time we can arrive at city i, after having gone sightseeing at exactly j different cities. For the base case of f(2, 0), this is equal to *Arrival*(2, 0) as defined above, while f(2, 1) = *Arrival*(2, $\mathbf{T_s}$).

To obtain the recurrence relation, consider a particular city i, where i > 1. You can either choose to go sightseeing there, or not. If you do, then that delays your departure time by $\mathbf{T_s}$ but adds 1 to the total number of cities in which you have gone sightseeing. This relationship can be captured as the following equation:

f(i, j) = min(*Arrival*(i, f(i - 1, j)), *Arrival*(i, f(i - 1, j - 1) + $\mathbf{T_s}$))

Once we have computed all possible f, the answer can be obtained by finding the maximum value of x such that f(**N**, x) ≤ $\mathbf{T_f}$. This solution runs in O($\mathbf{N}^2$) time.