Kick Start 2020 - Round B

Analysis: Bus Routes

We need to take the buses in order from 1 to **N**. Let the buses be B_1, B_2, ..., $B_{\mathbf{N}}$. Also, let us define a *good starting day* to be any day Y in the range [1..**D**] such that it is possible to start the journey on day Y and take all buses in the order from 1 to **N** before the end of day **D**. Note that we do not require Y to be a multiple of $\mathbf{X_1}$, so there may be some waiting time involved in the beginning of the journey.

For a fixed day Y, let us see if it is a good starting day or not. The best strategy would be to take bus B_1 as early as possible on or after day Y. This is because it would give us more days to take subsequent buses. Let us say we took bus B_1 on day D_1. Now the best strategy would be to take bus B_2 as early as possible on or after day D_1. Thus, if we took bus B_i on day D_i, it would be optimal to take bus B_{i+1} as early as possible on or after day D_i.

Now we need to find out what is the earliest possible day for Bucket to take bus B_i on or after a particular day K. Since bus B_i only runs on days that are multiples of $\mathbf{X_i}$, we need to find the smallest multiple of $\mathbf{X_i}$ greater than or equal to K. This can be calculated using the formula $\lceil K / \mathbf{X_i} \rceil * \mathbf{X_i}$. Thus if bus B_i is taken on day D_i, then it would be optimal to take bus B_{i+1} on day $D_{i+1} = \lceil D_i / \mathbf{X_{i+1}} \rceil * \mathbf{X_{i+1}}$. Thus, day Y is a good starting day if $D_{\mathbf{N}} \le \mathbf{D}$, and this question can be answered in O(**N**) time.

Test set 1

D can be at most 100, so we can find the latest good starting day by using the above approach for each day Y in the range [1..**D**]. The time complexity of this naive algorithm is O(**DN**).

Test set 2

Now **D** can be at most 10^{12}, so the naive algorithm would time out. Consider the largest good starting day P. Obviously, any day before P would be good as well because we can take the buses on the same days as if we started the journey on day P. Because of this observation, we can binary search on the range from 1 to **D** to find the largest good starting day P. The time complexity of the solution is O(**N** log **D**). Note that we can reduce the time complexity to O(**N** log(**D**/$\mathbf{X_1}$)) by restricting the search to multiples of $\mathbf{X_1}$ only.

Alternate solution

It is possible to solve the problem in linear time by working out the solution backwards. If we want to start our journey as late as possible, we should try to take the last bus $B_{\mathbf{N}}$ as late a possible, namely, on day $D_{\mathbf{N}}$, which is the largest multiple of $\mathbf{X_N}$, less than or equal to day **D**. Similarly, in order to be on time for the last bus on day $D_{\mathbf{N}}$, we have to take bus $B_{\mathbf{N}-1}$ no later than on day $D_{\mathbf{N}-1}$, which is the largest multiple of $\mathbf{X_{N-1}}$, less than or equal to $D_{\mathbf{N}}$. In general, bus B_i should be taken no later than on day D_i, which is the largest multiple of $\mathbf{X_i}$, less than or equal to D_{i+1}. The last calculated value D_1 is the answer to the problem.

Note that the largest multiple of $\mathbf{X_i}$ that occurs before a day L can be calculated in constant time as L - L mod $\mathbf{X_i}$. Therefore, the overall time complexity of this solution is O(**N**).