Code Jam 2008 - EMEA Semifinal

Analysis: Bus Stops

The problem can be solved by using dynamic programming and matrix multiplication for solving linear recursive sequences.

Consider a configuration of buses that are within a window of width P and advance the leftmost bus in such a way that all of the buses are still within a (shifted) window of width P.

Now we can define a state as the position of the buses within P units. For instance if we have P = 10 and K = 5 buses, then we have 252 possible states (10 choose 5). Let NS be the number of states.
To be sure we don't count the same state in different windows we can required that there always be a bus at the leftmost position in the window.

To advance the window of size P we move the leftmost bus. There is always a bus there. Now remember that the new state has to have a bus at the leftmost position, so the window might move to the right by more than one spot.

We compute all the possible state transitions. Let C be the matrix of state transitions having $C_{a,b}$ be the number of ways we can go from state a to state b.
For each state j, with the window in the position i we will have something like:

$$A[S_j][i+1] = C_{1,j}A[S_1][i] + C_{2,j}A[S_2][i] + \ldots + C_{NS,j}A[S_{NS}][i]$$

A[s][p] is the number of ways we can get into state s while having the window in position p.

This would be enough to solve the small input. For the large input we need to speed things up, so we are going to compute that linear recurrence of order NS by using matrix multiplication. Let's look instead at a much simpler example of a linear recurrence -- Fibonacci numbers.

$$F_N = F_{N-1} + F_{N-2}$$

This can be rewritten as:

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} * \begin{pmatrix} F_{N-2} \\ F_{N-1} \end{pmatrix} = \begin{pmatrix} F_{N-1} \\ F_N \end{pmatrix}$$

In a shorter form we have:

$$M * V_{N-1} = V_N$$

This says that to compute $V_N = (F_N, F_{N-1})$ from $V_{N-1} = (F_{N-2}, F_{N-1})$ we need to multiply with M. Now we can apply this recursively:

$$M^X * V_0 = V_X$$

M^X can be computed using an algorithm called *successive squaring* in time that is logarithmic in X.