USACO NOV13 Problem 'empty' Analysis

by Bruce Merry

This problem is inspired by closed hash tables with linear probing, which use the same algorithm as the cows to assign elements to positions in the hash table.

The problem gives us a hint by telling us that the answer does not depend on the order in which cows enter the barn: we don't need to know where each cow ends up, only which stalls are empty. Rather than adding cows one at a time, let's do things in a different order that yields the same result.

First, put each cow in her preferred stall (this may cause some overcrowding, but fortunately cows are very easy-going). Then, sweep from low stalls to high stalls, and whenever a stall with more than one cow is found, move all but one of them to the next stall along - this is what those cows would do themselves anyway. On reaching stall N-1, one might then need to move some cows back to stall 0, and continue the process a second time. Two passes is sufficient to reach a stable situation: after the first pass, the only cows that need to be moved along are those brought from stall N-1, and if they are moved all the way back to stall N-1 it implies that there were more cows than stalls. Once this process is complete, one simply finds the first stall with no cows. Below is Rumen Hristov's solution to the problem. Note that it's not even necessary to modify the preference array

```
#include <cstdio>

int a[1 << 22];

int main() {
        freopen ("empty.in", "r", stdin );
        freopen ("empty.out", "w", stdout);
        int n, k, x, y;
        long long i, A, B;
        int sum = 0;

        scanf ("%d%d", &n, &k);
        while (k --) {
                scanf("%d%d%lld%lld", &x, &y, &A, &B);

                for (i = 1; i <= y; i++)
                        a[(A * i + B) % n] += x;
        }

        for (i = 0; i < n; i++) {
                sum += a[i];
                if (sum > 0) sum --;
        }

        for (i = 0; i < n; i++) {
                sum += a[i];
                if (sum > 0) sum --;
                else {
                        printf ("%lld\n", i);
                        break;
                }
        }
}
```