USACO JAN10 Problem 'hayturn' Analysis

by John Pardon

This is a dynamic programming problem. One easy way to see this is to observe that if the cows are at haybale number M, then what happened earlier (i.e. who got which of the haybales 1..M-1) has no effect on the subsequent choices made by the cows. Let:

F(m) = the maximum amount of hay that a cow can eat supposing it is her turn and the first bale she may pick is bale number m.

S(m) = the maximum amount of hay that a cow can eat supposing it is NOT her turn and the first bale the OTHER cow may pick is bale number m.

Now by some easy reasoning, we can find a DP recurrence. Suppose it is Bessie's turn and she can start picking bales at #m. She has two choices: if she skips bale #m, then she will eat a total of F(m+1) hay. If she eats bale #m, then she will eat a total of S(m+1)+W_m hay. Using this reasoning, we have the following:

If W_m+S(m+1) >= F(m+1), then F(m)=S(m+1)+W_m and S(m)=F(m+1)

If W_m+S(m+1) < F(m+1), then F(m)=F(m+1) and S(m)=S(m+1)

Now one just uses dynamic programming to calculate F(1) and S(1) starting at the end of the haybale line and working backwards. A solution follows:

```
#include <cstdio>

using namespace std;

int N;
int w[700000];

int main(void)
{
        FILE *inFile = fopen("hayturn.in", "rt");
        fscanf(inFile, "%i", &N);
        for(int i = 0 ; i < N ; i++)
                fscanf(inFile, "%i", &w[i]);
        fclose(inFile);

        long long A = 0;
        long long B = 0;
        for(int i = N - 1 ; i >= 0 ; i--)
        {
                if(B + w[i] >= A)
                {
                        long long temp = A;
                        A = B + w[i];
                        B = temp;
                }
        }

        FILE *outFile = fopen("hayturn.out", "wt");
        fprintf(outFile, "%lld %lld\n", A, B);
        fclose(outFile);
}
```