

(a) Suppose that  $M = 10$ ,  $\{N_1, N_2, N_3\} = \{1, 4, 1\}$ , and  $\{S_1, S_2, S_3\} = \{20, 1, 20\}$ . Then the optimal plan would be  $[NY, NY, NY]$ , while this greedy algorithm would return  $[NY, SF, NY]$ .

(b) Suppose that  $M = 10$ ,  $\{N_1, N_2, N_3, N_4\} = \{1, 100, 1, 100\}$ , and  $\{S_1, S_2, S_3, S_4\} = \{100, 1, 100, 1\}$ .

Explanation: The plan  $[NY, SF, NY, SF]$  has cost 34, and it moves three times. Any other plan pays at least 100, and so is not optimal.

(c) The basic observation is: The optimal plan either ends in NY, or in SF. If it ends in NY, it will pay  $N_n$  plus one of the following two quantities:

- The cost of the optimal plan on  $n - 1$  months, ending in NY, or
- The cost of the optimal plan on  $n - 1$  months, ending in SF, plus a moving cost of  $M$ .

An analogous observation holds if the optimal plan ends in SF. Thus, if  $OPT_N(j)$  denotes the minimum cost of a plan on months  $1, \dots, j$  ending in NY, and  $OPT_S(j)$  denotes the minimum cost of a plan on months  $1, \dots, j$  ending in SF, then

$$OPT_N(n) = N_n + \min(OPT_N(n-1), M + OPT_S(n-1))$$

$$OPT_S(n) = S_n + \min(OPT_S(n-1), M + OPT_N(n-1))$$

This can be translated directly into an algorithm:

```

 $OPT_N(0) = OPT_S(0) = 0$ 
For  $i = 1, \dots, n$ 
   $OPT_N(i) = N_i + \min(OPT_N(i-1), M + OPT_S(i-1))$ 
   $OPT_S(i) = S_i + \min(OPT_S(i-1), M + OPT_N(i-1))$ 
End
Return the smaller of  $OPT_N(n)$  and  $OPT_S(n)$ 

```

The algorithm has  $n$  iterations, and each takes constant time. Thus the running time is  $O(n)$ .