

USACO NOV08 Problem 'buyhay' Analysis

by Neal Wu

This problem is a typical knapsack problem and can be solved using dynamic programming. We keep an array called `best []` such that `best [h]` is the minimum cost possible to obtain exactly `h` pounds of hay. For each supplier given, we do a simple traversal from 0 to `H` to update the values in the array. In particular, if `(best [i] + the cost of one package)` is better than `best [i + the weight of one package]`, we update this value accordingly.

At the end, our answer is simply the minimum cost of any value in the array that is at position `H` or after. Since our algorithm does `H` iterations for each of the `N` suppliers, our overall runtime is $O(NH)$.

The following is a sample solution:

```
#include <stdio>
#include <cstring>
using namespace std;

FILE *fout = fopen ("buyhay.out", "w");
FILE *fin = fopen ("buyhay.in", "r");

const int INF = 1000000000;
const int MAXH = 60005;

int N, H, P, C;
int best [MAXH];
int ans = INF;

int main ()
{
    // initialize to infinity
    memset (best, 63, sizeof (best));
    best [0] = 0;

    fscanf (fin, "%d %d", &N, &H);

    while (N--)
    {
        fscanf (fin, "%d %d\n", &P, &C);

        // update values in the array; only necessary to go up to H
        for (int i = 0; i < H; i++)
            if (best [i] + C < best [i + P])
                best [i + P] = best [i] + C;
    }

    // find best cost for any amount that is at least H
    for (int i = H; i < MAXH; i++)
        if (best [i] < ans)
            ans = best [i];

    fprintf (fout, "%d\n", ans);

    return 0;
}
```