

USACO FEB14 Problem 'dec' Analysis

by William Hu and Steven Hao

A very simple solution would be to try all $N!$ assignments for the cows to the events. Unfortunately, this is far too slow for $N = 20$. However, N is fairly small, so an exponential-time solution is likely to be feasible.

To improve the complexity, we instead try a different approach. For every subset S , we use dynamic programming to calculate the maximum number of points that can be earned if the first $|S|$ cows that participate in the events are the cows in S .

The total number of states is $O(2^N)$. At each state, there are at most N ways to choose the cow who competed in the last event. Now, we need to calculate the resulting score once the judges award bonuses. We do this as follows:

Let $\text{award}(A, X)$ be the score of the cows after the judges award the points after the cows have earned A points and they have participated in X events. We can greedily calculate $\text{award}(A, X)$ by sorting the awards at each event by their point requirements. If we calculate the award for each event in the $O(N)$ transitions, the total time complexity would be $O(2^N N B)$. However, we can do better by noting the award function is nondecreasing in terms of A . So, we only have to calculate the award for the maximal transition out of the $O(N)$ transitions. This makes the time complexity $O(2^N N)$ for all transitions and $O(2^N B)$ to calculate the award, making the total runtime $O(2^N (N + B))$.

Below is a C++ solution:

```
#include <cstdio>
#include <vector>
#include <utility>
#include <algorithm>

using namespace std;
typedef pair<int, int> pii;

const int MAXN = 20;

int N, B;
int dp[1 << MAXN];
vector<pii> bonus[MAXN];
int skill[MAXN][MAXN];

int award (int score, int event)
{
    //award new bonuses
    int siz = bonus[event].size();
    for (int i = 0; i < siz; i++)
    {
        if (score < bonus[event][i].first)
            break;
        score += bonus[event][i].second;
    }
    return score;
}

int main()
{
    freopen("dec.in", "r", stdin);
    freopen("dec.out", "w", stdout);
```

```

scanf("%d %d", &N, &B);
for (int i = 0; i < B; i++)
{
    int k, p, b;
    scanf("%d %d %d", &k, &p, &b);
    --k;
    bonus[k].push_back(pii(p, b)); //the points and bonus
}
for (int i = 0; i < N; i++)
    sort(bonus[i].begin(), bonus[i].end()); //sort bonuses greedily
for (int i = 0; i < N; i++)
{
    for (int j = 0; j < N; j++)
        scanf("%d", &skill[i][j]); //cow, event
}
for (int i = 1; i < (1 << N); i++)
{
    int b = __builtin_popcount(i);
    for (int j = 0; j < N; j++)
    {
        if (i & (1 << j))
        {
            int x = dp[i ^ (1 << j)] + skill[j][b - 1];
            if (dp[i] < x)
                dp[i] = x;
        }
    }
    dp[i] = award(dp[i], b - 1);
}
printf("%d\n", dp[(1 << N) - 1]);
return 0;
}

```