SPOJ WACHOVIA

Wachovia Bank

Prof. Edson Alves – UnB/FGA

Problema

Danilo Gheyi is a renowned bank robber. He is known worldwide for accomplishing the most profitable bank robbery, in Fortaleza, Ceará. Danilo and his friends dug a tunnel to get into the main chest. There were some bags, with different amounts of money or jewelry and weight. They had to leave about 50% of the total value, since the truck couldn't carry all the bags.

Danilo wasn't caught, and to show that he can do it all again, he is planning a robbery to one of the safer banks in USA – the Wachovia Bank. He wants your help to maximize the amount stolen, avoiding a huge loss as it happened in Fortaleza.

Write a program that, given the maximum weight the truck is able to carry and the information about each bag in the bank, determine the maximum value that Danilo can steal.

Entrada e saída

Input

The input consists of several instances. There is an integer N ($1 \leq N \leq 200$) in the first line; it stands for the number of instances. The first line of each instance contains two integers, K and M ($8 \leq K \leq 1000$ and $1 \leq M \leq 50$) representing, respectively, the maximum weight the truck can handle and the amount of bags in the bank. The next M lines describe each bag with two integers A and B ($8 \leq A \leq 200$ and $1 \leq B \leq 25$): the weight and the value of the bag, respectively.

Output

For each instance output a sentence "`Hey stupid robber, you can get` P`.`", and P represents the maximum value Danilo can steal.

Exemplo de entradas e saídas

Sample Input

```
3
34 5
178 12
30 1
13 7
34 8
87 6
900 1
900 25
100 10
27 16
131 9
132 17
6 5
6 23
56 21
100 25
1 25
25 25
100 2
```

Sample Output

```
Hey stupid robber, you can get 8.
Hey stupid robber, you can get 25.
Hey stupid robber, you can get 99.
```

Solução $O(KM)$

- O problema apresentado pode ser modelado como um problema da mochila binária
- O caminhão é a mochila do problema, com capacidade K
- Os elementos da mochila binária são os pertences a serem subtraídos do banco
- Os pesos e os valores dos elementos são A e B, respectivamente
- Estabelecida estas correspondências, basta aplicar a solução de programação dinâmica para o problema da mochila, sem alteração alguma
- Portanto, a complexidade da solução será $O(KM)$

Solução $O(KM)$

```cpp
#include <bits/stdc++.h>

using namespace std;
using ii = pair<int, int>;
using ll = long long;

const int MAXN { 60 }, MAXM { 1010 };

ll st[MAXN][MAXM];

ll dp(int i, int m, int M, const vector<ii>& cs)
{
    if (i < 0)
        return 0;

    if (st[i][m] != -1)
        return st[i][m];

    auto res = dp(i - 1, m, M, cs);
    auto w = cs[i].first, v = cs[i].second;
```

Solução $O(KM)$

```
22      if (w <= m)
23          res = max(res, dp(i - 1, m - w, M, cs) + v);
24
25      st[i][m] = res;
26      return res;
27  }
28
29  ll knapsack(int M, const vector<ii>& cs)
30  {
31      memset(st, -1, sizeof st);
32
33      return dp((int) cs.size() - 1, M, M, cs);
34  }
35
36  int main()
37  {
38      ios::sync_with_stdio(false);
39
40      int T;
41      cin >> T;
```

Solução $O(KM)$

```
43      while (T--)
44      {
45          int M, N;
46          cin >> M >> N;
47
48          vector<ii> cs(N);
49
50          for (int i = 0; i < N; ++i)
51              cin >> cs[i].first >> cs[i].second;
52
53          cout << "Hey stupid robber, you can get " << knapsack(M, cs) << ".\n";
54      }
55
56      return 0;
57  }
```