

20230525 第四次作业

题目 1: Bone collector(hdu 2602)

题目描述:

骨头收集者带着体积为 V 的背包去捡 N 个骨头, 已知每块骨头的体积和价值, 求能装进背包的最大价值。

输入: 第1行输入测试数量; 后面每3行为一个测试, 其中第1行输入骨头数量 N 和背包体积 V 。第2行输入每块骨头的价值 w , 第3行输入每块骨头的体积。

输出: 最大价值。

$N \leq 1000, V \leq 1000$

题解:

非常经典的背包问题。

唯一需要注意的是在每个案例开始前要初始化一次 dp 数组。

时间复杂度 $O(nV)$, 空间复杂度 $O(n^2)$ 。

38564274	2023-05-25 17:41:04	Accepted	2602	140MS	5760K	705 B	C++	SevenChen
----------	---------------------	----------	------	-------	-------	-------	-----	-----------

```

#include<iostream>
#include<algorithm>
#define N 1005
using namespace std;
int val[N], siz[N], dp[N][N];
int main()
{
    ios::sync_with_stdio(false);
    int t;
    cin >> t;
    while (t--) {
        int n, V;
        cin >> n >> V;
        memset(dp, 0, sizeof(dp));
        for (int i = 1; i <= n; i++)
            cin >> val[i];
        for (int i = 1; i <= n; i++)
            cin >> siz[i];
        for (int i = 1; i <= n; i++)
            for (int j = 0; j <= V; j++) {
                if (siz[i] > j)
                    dp[i][j] = dp[i - 1][j];
                else
                    dp[i][j] = max(dp[i - 1][j], dp[i - 1][j - siz[i]] + val[i]);
            }
        cout << dp[n][V] << endl;
    }
    return 0;
}

```

题目 2: ACboy needs your help (hdu 1712)

题目描述:

ACboy这学期可以选 N 门课, 他只想学 M 天。每门课的学分不同, 问这 N 天如何安排 N 门课, 才能得到最多学分?

输入: 有多个测试。每个测试的第1行输入 N 和 M 。后面有 N 行, 每行输入 M 个数字, 表示一个矩阵 $A[i][j]$, $1 \leq i \leq N \leq 100, 1 \leq j \leq M < 100$, 表示第 i 门课学 j 天能得到 $A[i][j]$ 学分。若 $N = M = 0$, 表示测试结束。

输出: 对每个测试, 输出最多学分。

题解:

可以看作多个背包问题。

N 门课看作 N 件物品, M 天看作 M 容量。

需要考虑每件物品有 N 种体积, 每种体积对应一种 val , 则其实共有 $N * N$ 种物品, 且每种物品只能同时存在一种体积。

若用二维数组，对于同一种物品的不同体积并不方便处理，考虑用滚动数组，依次考虑第 i 种物品在容量为 j 的条件下 k 体积的 val 。

时间复杂度 $O(nm^2)$ 。

38565388	2023-05-25 22:34:21	Accepted	1712	280MS	1896K	554 B	C++	SevenChen
----------	---------------------	----------	------	-------	-------	-------	-----	-----------

C++

```
#include<iostream>
#include<algorithm>
using namespace std;
int siz[105][105], val[105][105], dp[105];
int main()
{
    ios::sync_with_stdio(false);
    int n, m;
    while (cin >> n >> m && n && m) {
        for (int i = 1; i <= n; i++)
            for (int j = 1; j <= m; j++) {
                cin >> val[i][j];
                siz[i][j] = j;
            }
        fill(dp, dp + 105, 0);
        for (int i = 1; i <= n; i++)
            for (int j = m; j >= 0; j--)
                for (int k = 1; k <= m; k++)
                    if (j >= siz[i][k])
                        dp[j] = max(dp[j], dp[j -
siz[i][k]] + val[i][k]);
        cout << dp[m] << "\n";
    }
    return 0;
}
```

题目 3：手机号码(洛谷 P4124)

题目描述：

手机号码一定是 11 位数，前不含前导的 0。

需要检测的号码特征有两个：号码中要出现至少 3 个相邻的相同数字；号码中不能同时出现 8 和 4。

工具接收两个数 L 和 R ，自动统计出 $[L, R]$ 区间内所有满足条件的号码数量。 L 和 R 也是 11 位的手机号码。

$10^{10} \leq L \leq R < 10^{11}$

题解：

数位 dp 。

`dp[p][a][b][if3][s][if4][if8]` 依次表示第 p 位，第 $p+1$ 位为 a ，第 $p+2$ 位为 b ，是否出现三个连续相同数字，是否小于 n ，是否出现 4，是否出现 8。



WilliamXI

所属题目	P4124 [CQOI2016]手机号码
评测状态	Accepted
评测分数	100
提交时间	2023-05-27 20:52:00

```

#include<iostream>
#include<vector>
#include<string.h>
#define ll long long
using namespace std;
vector<int> nums;
ll dp[11][11][11][2][2][2][2];
ll f(int p, int a, int b, bool if3, bool s, bool if4, bool if8) {
    if (if4 && if8)
        return 0;
    if (!p)
        return if3;
    if (dp[p][a][b][if3][s][if4][if8] != -1)
        return dp[p][a][b][if3][s][if4][if8];
    ll res = 0;
    int lim = s ? 9 : nums[p-1];
    for (int i = 0; i <= lim; i++)
        res += f(p - 1, i, a, if3 || (i == b && i == a), s || (i <
lim), if4 || i == 4, if8 || i == 8);
    return dp[p][a][b][if3][s][if4][if8] = res;
}
ll func(ll n) {
    if (n < 1e10)
        return 0;
    nums.clear();
    while (n) {
        nums.push_back(n % 10);
        n /= 10;
    }
    memset(dp, -1, sizeof(dp));
    ll res = 0;
    for (int i = 1; i <= nums.back(); i++)
        res += f(10, i, 0, 0, i < nums.back(), i == 4, i == 8);
    return res;
}
int main()
{
    ios::sync_with_stdio(false);
    ll L, R;
    cin >> L >> R;
    cout << func(R) - func(L - 1) << "\n";
    return 0;
}

```

题目 4：没有上司的舞会（洛谷 P1352）

题目描述：

n 个职员，父节点表示子节点的直接上司。

职员 i 快乐指数 r_i ，职员与其直接上司不会同时出现。

求最大快乐指数。

输入：第 1 行输入一个整数 n 。第 2 $\sim n + 1$ 行中，每行输入一个整数，第 $i + 1$ 行的整数标识职员 i 的快乐指数 r_i 。第 $n + 2 \sim 2n$ 行中，每行输入两个整数 l 和 k ，代表 k 是 l 的直接上司。

输出：输出一行一个整数代表最大的快乐指数。

$1 \leq n \leq 6 \times 10^3$ ， $-128 \leq r_i \leq 127$ ， $1 \leq l, k \leq n$

题解：

设 $f(i, 0/1)$ 表示以 i 为根的子树中， i 是否参加时的最大快乐值，则我们可以列出状态转移方程：

$$f(u, 0) = \sum_{v \in \text{son}(u)} \max(f(v, 0), f(v, 1))$$

$$f(u, 1) = \sum_{v \in \text{son}(u)} f(v, 0) + \text{val}_u$$

其中 $\text{son}(u)$ 表示以 u 为父节点的子节点集合， val_u 表示节点 u 的快乐值。

由于叶子节点有 $f(i, 1) = \text{val}_i$ ， $f(i, 0) = 0$ ，所以最后的答案应该为 $\max(f(i, 1), f(i, 0))$ 。

时间复杂度为 $O(n)$ ，空间复杂度为 $O(n)$ 。



WilliamXI

所属题目

P1352 没有上司的舞会

评测状态

Accepted

评测分数

100

提交时间

2023-05-27 21:33:33

```

#include<iostream>
#include<vector>
#include<algorithm>
using namespace std;
const int N = 6005;
int n, val[N], dp[N][2];
vector<int> g[N];
void dfs(int u) {
    dp[u][1] = val[u];
    dp[u][0] = 0;
    for (int i = 0; i < g[u].size(); i++) {
        dfs(g[u][i]);
        dp[u][1] += dp[g[u][i]][0];
        dp[u][0] += max(dp[g[u][i]][0], dp[g[u][i]][1]);
    }
}
int main()
{
    cin >> n;
    for (int i = 1; i <= n; i++)
        cin >> val[i];
    bool findroot[N] = { 0 };
    for (int i = 1; i < n; i++) {
        int l, k;
        cin >> l >> k;
        g[k].push_back(l);
        findroot[l] = 1;
    }
    int root;
    for (int i = 1; i <= n; i++)
        if (!findroot[i])
            root = i;
    dfs(root);
    cout << max(dp[root][0], dp[root][1]) << "\n";
    return 0;
}

```