20230525 第四次作业实验报告

学号：2022212408 姓名：胡宇杭

2023 年 9 月 20 日

提交结果

Lieserl	D	Accepted	40	1.1	C++	2 min ago
Lieserl	C	Accepted	40	0.8	C++	10 hr ago
Lieserl	B	Accepted	296	1.8	C++	15 hr ago
Lieserl	A	Accepted	62	1.8	C++	16 hr ago

图 1: 提交结果

1 HDU-2602 Bone Collector

- 时间限制：1000 ms
- 空间限制：32768 KB

1.1 题目

1.1.1 问题描述

Bone Collector 带着一容量为 V 的背包，一共有 N 块骨头。对每一次测试，已知每块骨头的体积和价值，问能装下骨头的最大价值?

1.1.2 子任务

对于所有测试数据，满足 $N \leq 1000$，$V \leq 1000$。

1.2 题解

1.2.1 解法一 (100 分)

根据题意,考虑使用 dp 求解(都~~学期末了连 01 背包都不知道的可以洗洗睡了~~)。状态转移方程如下:

$$f(i,j) = max(f(i-1,j), f(i-1,j-w_i) + v_i)$$

- 时间复杂度: $O(nm)$
- 空间复杂度: $O(n)$

1.2.2 C++ 代码实现

HDU-2602 Bone Collector

```cpp
#include <iostream>
#include <algorithm>
#include <cstring>
using namespace std;
const int maxn = 1005;

int t, n, c, w[maxn], v[maxn], dp[maxn];

int main()
{
    cin >> t;
    while (t--)
    {
        cin >> n >> c;

        for (int i = 1; i <= n; i++)
            cin >> v[i];

        for (int i = 1; i <= n; i++)
```

```
20              cin >> w[i];
21
22          memset(dp, 0, sizeof(dp));
23
24          for (int i = 1; i <= n; i++)
25              for (int j = c; j >= w[i]; j--)
26                  dp[j] = max(dp[j], dp[j - w[i]] + v[i]);
27
28          cout << dp[c] << endl;
29      }
30
31      // system("pause");
32
33      return 0;
34  }
```

2 HDU-1712 ACboy needs your help

- 时间限制：1000 ms
- 空间限制：32768 KB

2.1 题目

2.1.1 问题描述

ACboy 这学期可以选择 N 门课程，~~但 ACboy 是个摆子~~，所以这学期他只学 M 天。对于课程 i $(1 \leq i \leq N)$，ACboy 学习 j $(1 \leq j \leq M)$ 天可以获得学分 A_{ij}，问 ACboy 怎么安排可以获得最大学分?

2.1.2 子任务

对于所有测试数据，满足 $N \leq 100$，$M \leq 100$。

2.2 题解

2.2.1 解法一 (100 分)

与题一不同，每个课程的价值不在固定，考虑使用分组背包解决。具体来说即是在 01 背包的基础上再加一层循环遍历第 i 门课程学习 j 天的最优解。状态转移方程如下：

$$dp(i,j) = max(dp(i-1,j), dp(i-1,j-k) + a_{ik})$$

其中 i 代表当前选择的课程，j 代表当前剩余可学习天数，k 代表当前课程的学习天数。

- 时间复杂度：$O(nm^2)$
- 空间复杂度：$O(nm)$

2.2.2 C++ 代码实现

HDU-1712 ACboy needs your help

```cpp
#include <iostream>
#include <algorithm>
#include <cstring>
using namespace std;
const int maxn = 105;

int n, m, a[maxn][maxn], dp[maxn];

int main()
{
    while ((cin >> n >> m) && n && m)
    {
        for (int i = 1; i <= n; i++)
            for (int j = 1; j <= m; j++)
                cin >> a[i][j];

        memset(dp, 0, sizeof(dp));

        for (int i = 1; i <= n; i++)
            for (int j = m; j >= 0; j--)
```

```
21                 for (int k = 1; k <= m; k++)
22                     if (j >= k)
23                         dp[j] = max(dp[j], dp[j - k] + a[i][k]);
24
25         cout << dp[m] << endl;
26     }
27
28     // system("pause");
29
30     return 0;
31 }
```

3 洛谷-P4124 手机号码

3.1 题目

- 时间限制：1.0 s
- 空间限制：250 MB

3.1.1 问题描述

现需要统计号码段中满足某些特征的手机号的数量，特征如下：

- 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 4 和 8。

手机号码一定是 11 位数，前不含前导 0，给定 L，R，统计 $[L, R]$ 区间内所有满足条件的号码数量。

3.1.2 子任务

对于所有测试数据，满足 $10^10 \leq L \leq R \leq 10^11$。

3.2 题解

3.2.1 解法一 (100 分)

数位 dp，暴力枚举!。假设 $dp(pos, pre_1, pre_2, 0/1, 0/1, 0/1)$，其中，$pos$ 代表当前位数，pre_1、pre_2 分别代表 $pos+1$、$pos+2$ 位的数，三个 0/1 分别代表是否出现过 3 个连续且相同的数、是否出现过 4、是否出现过 8，按照题目条件确定状态如何转移即可。由于 dp 代码实现晦涩难懂（懒），这里采用 dfs 记忆化搜索的形式。

3.2.2 C++ 代码实现

洛谷-P4124 手机号码

```cpp
#include <iostream>
#include <algorithm>
#include <cstring>
using namespace std;
typedef long long ll;

ll l, r, num[12], dp[11][11][11][2][2][2];

ll dfs(int pos, int pre_1, int pre_2, bool state, bool flag, bool _4, bool _8)
{
    if (_4 && _8)
        return 0;
    if (pos <= 0)
        return state;
    if (!flag && dp[pos][pre_1][pre_2][state][_4][_8] != -1)
        return dp[pos][pre_1][pre_2][state][_4][_8];

    int lim = flag ? num[pos] : 9;
    ll ret = 0;

    for (int i = 0; i <= lim; i++)
        ret += dfs(pos - 1, i, pre_1, state || (i == pre_2 && i == pre_1), flag && (i == num
            [pos]), _4 || (i == 4), _8 || (i == 8));

    if (!flag)
        dp[pos][pre_1][pre_2][state][_4][_8] = ret;
```

```
26
27      return ret;
28  }
29
30  ll solve(ll x)
31  {
32      if (x < 1e10)
33          return 0;
34
35      int len = 0;
36      ll ret = 0;
37      memset(dp, -1, sizeof(dp));
38
39      while (x)
40      {
41          num[++len] = x % 10;
42          x /= 10;
43      }
44
45      for (int i = 1; i <= num[len]; i++)
46          ret += dfs(10, i, 0, 0, i == num[len], i == 4, i == 8);
47
48      return ret;
49  }
50
51  int main()
52  {
53      cin >> l >> r;
54
55      cout << solve(r) - solve(l - 1) << endl;
56
57      // system("pause");
58
59      return 0;
60  }
```

4 洛谷-P1352 没有上司的舞会

4.1 题目

- 时间限制：1000 ms
- 空间限制：134144 KB

4.1.1 问题描述

某大学有 n 个职员，他们之间有从属关系，除校长以外每个职员都有直接上司。现有一周年庆宴会，宴会邀请编号为 i 的职员会增加快乐指数 r_i，但是一个职员和他的直接上司不会同时来参加宴会。问邀请哪些职员可以使宴会的快乐指数最大?

4.1.2 子任务

对于所有测试数据，满足 $1 \leq n \leq 6 * 10^3$，$-128 \leq r_i \leq 127$，$1 \leq l, k \leq n$，且给出的关系一定是一棵树。

4.2 题解

4.2.1 解法一 (100 分)

该题可以抽象成一棵以校长为根的树，对于这种情况需使用树形 dp 求解。假设 $dp(i, 0/1)$ 表示以 i 为根节点的子树在不邀请 / 邀请 i 时可达到的最大快乐指数，则状态转移方程为:

$$dp(i,1) = \sum_{j=son[0]}^{sizeof(son)} dp(j,0)$$

$$dp(i,0) = \sum_{j=son[0]}^{sizeof(son)} max(dp(j,0), dp(j,1))$$

同时，定义 $bool$ 数组 vis 记录节点 i 是否有父亲从而寻找根节点，从根节点开始 dfs 即可。

4.2.2 C++ 代码实现

洛谷-P1352 没有上司的舞会

```cpp
#include <iostream>
#include <algorithm>
#include <cstring>
#include <vector>
using namespace std;

const int maxn = 6e3 + 10;

vector<int> edge[maxn];
int n, r[maxn], l, k, dp[maxn][2];
bool vis[maxn];

void dfs(int x)
{
    dp[x][0] = 0;
    dp[x][1] = r[x];

    for (auto &i : edge[x])
    {
        dfs(i);
        dp[x][0] += max(dp[i][1], dp[i][0]);
        dp[x][1] += dp[i][0];
    }
}

int main()
{
    cin >> n;
    memset(vis, false, sizeof(vis));

    for (int i = 1; i <= n; i++)
        cin >> r[i];

    for (int i = 1; i < n; i++)
    {
        cin >> l >> k;
        edge[k].push_back(l);
        vis[l] = true;
```

```
39      }
40
41      int rt = 0;
42      while (vis[++rt])
43      {
44      }
45
46      dfs(rt);
47
48      cout << max(dp[rt][0], dp[rt][1]) << endl;
49
50      // system("pause");
51
52      return 0;
53  }
```