

搜索剪枝

人员

柳力玮、田心一、李瑞涵、李知朔、纪博涵、蒋叔璋、初锦阳 到课, 苑钊、高健桓 线上

上周作业检查

上周作业链接: <https://cppoj.kids123code.com/contest/1912>



#	用户名	姓名	编程分	时间	A	B	C	D	E	F
1	liuliwei	柳力玮	600	549	100	100	100	100	100	100
2	tianxinyi	田心一	557	279	100	100	100	100	57	100
3	zhaoshufan	赵书梵	553	148	100	100	100	100	53	100
4	yuanzhao	苑钊	500	203	100	100	100	100		100
5	gaojianhuan	高健桓	500	230	100	100	100	100		100
6	liruihan	李瑞涵	500	248	100	100	100	100		100
7	lizishuo	李知朔	446	180	100	100	100	100	46	
8	jiangchengen1	蒋承恩	435	598	100	100	100	100	35	
9	liuchenxi	刘宸熙	400	139	100	100	100	100		
10	jiangshuzhang	蒋叔璋	300	71	100	0	100	100		
11	jibohan	纪博涵	300	202	100	100	100	0		
12	chujin yang	初锦阳	240	180	100	100		40		

本周作业

<https://cppoj.kids123code.com/contest/2022> (课上讲了 A ~ C 题, 课后作业是 D 题)

课堂表现

今天的 C 题本质上跟 A 题是一样的, 同学们普遍 A 题都做的比较好, 但是 C 题都做的不太好, 所以 C 题需要同学们课下再好好复习复习。

课堂内容

[GESP202403 六级] 游戏 (上周作业)

dfs(n): 凑出来 n 这个数一共有多少种可能的方案

dfs(n) 的值是由 $\text{dfs}(n-a) + \text{dfs}(n-b)$

把搜的方案数记到数组中即可

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

using namespace std;
```

```
const int maxn = 2e5 + 5;
const int mod = 1e9 + 7;
int f[maxn];
int a, b, c;

int dfs(int n) {
    if (n <= c) return 1;
    if (f[n]) return f[n];
    f[n] = (dfs(n-a) + dfs(n-b)) % mod;
    return f[n];
}

int main()
{
    int n; cin >> n >> a >> b >> c;
    cout << dfs(n) << endl;
    return 0;
}
```

B3624 猫粮规划

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

using namespace std;

const int maxn = 40 + 5;
int w[maxn];
int n, l, r;
int res = 0;

void dfs(int u, int sum) {
    if (sum > r) return;

    if (u == n+1) {
        if (sum>=l && sum<=r) ++res;
        return;
    }

    dfs(u+1, sum);
    dfs(u+1, sum+w[u]);
}

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

    sort(w+1, w+n+1), reverse(w+1, w+n+1);
    dfs(1, 0);
}
```

```

cout << res << endl;
return 0;
}

```

P2383 狗哥玩木棒

凑出 4 个 target 即可, 用 4 个数组代表 4 条边边长已经凑了多少

剪枝策略:

1. 保证每条边长都不能超过 target 即可
2. 已经成功之后, 不需要再继续进行剩余的搜索

```

#include <bits/stdc++.h>

using namespace std;

const int maxn = 1000 + 5;
int w[maxn], f[5];
int n, target;
bool flag;

void dfs(int u) {
    if (flag) return;
    if (u == n+1) { flag = true; return; }

    for (int i = 1; i <= 4; ++i) {
        if (f[i]+w[u] <= target) {
            f[i] += w[u];
            dfs(u+1);
            f[i] -= w[u];
        }
    }
}

void solve() {
    cin >> n;
    target = 0;
    for (int i = 1; i <= n; ++i) cin >> w[i], target += w[i];
    if (target%4) { cout << "no" << endl; return; }

    target /= 4;
    sort(w+1, w+n+1), reverse(w+1, w+n+1);
    flag = false;
    dfs(1);
    cout << (flag ? "yes" : "no") << endl;
}

int main()
{
    int T; cin >> T;

```

```

while (T --) solve();
return 0;
}

```

B4219 [常州市赛 2023] 数学作业

剪枝策略:

1. 斐波那契数列降序排序, 确保可以早进行剪枝
2. 后缀和维护, 若当前结果加上后面所有数的和都凑不够, 应该减掉
3. 当前的和已经超过目标数了, 也不需要往后搜了

```

#include <bits/stdc++.h>

using namespace std;

typedef long long LL;
const int maxn = 1e5 + 5;
const LL limit = 1e13 + 5;
LL w[maxn], x, suf[maxn];
int n, res = 0;

void dfs(int u, LL sum) {
    if (sum > x) return;
    if (sum+suf[u] < x) return;
    if (u == n+1) {
        if (sum == x) ++res;
        return;
    }
    dfs(u+1, sum);
    dfs(u+1, sum+w[u]);
}

int main()
{
    w[1] = 1, w[2] = 2;
    for (int i = 3; ; ++i) {
        w[i] = w[i-1] + w[i-2];
        if (w[i] >= limit) { n = i; break; }
    }

    cin >> x;
    reverse(w+1, w+n+1);
    suf[n] = w[n];
    for (int i = n-1; i >= 1; --i) suf[i] = suf[i+1] + w[i];

    dfs(1, 0);

    cout << res << endl;
}

```

```
    return 0;  
}
```