

# 剪枝

## 人员

秦显森、刘闯速、赵熙羽、牛同泽、高健桓、陈洛冉、程梓豪、孙靖轲、杨瑾硕、于子珈 到课, 牟茗 线上

## 上周作业检查

https://www.luogu.com.cn/contest/250020

2025-0601周日10:30

报名

编辑比赛

题目数5 | 报名人数13

比赛说明 | 题目列表 | 排行榜

名次	参赛者	总分	A	B	C	D	E
#1	秦显森	500 (7.19d)	100 (689ms)	100 (40ms)	100 (2.00h)	100 (2.08h)	100 (7.02d)
#2	赵熙羽	500 (15.92d)	100 (4.40d)	100 (32ms)	100 (2.25h)	100 (4.41d)	100 (7.02d)
#3	高健桓	475 (12.99d)	100 (675ms)	100 (31ms)	100 (1.95h)	100 (5.96d)	75 (6.95d)
#4	牛同泽	450 (34.50d)	100 (6.86d)	100 (6.89d)	100 (6.92d)	100 (6.92d)	50 (6.91d)
#5	于子珈	430 (27.13d)	100 (684ms)	100 (6.95d)	100 (6.95d)	100 (6.95d)	30 (6.27d)
#6	孙靖轲	425 (14.26d)	100 (735ms)	100 (23.47h)	100 (6.12d)	100 (6.14d)	25 (1.02d)
#7	谢亚锴	400 (5.35d)	100 (681ms)	100 (30ms)	100 (2.00h)	100 (5.26d)	
#8	陈洛冉	400 (7.08d)	100 (665ms)	100 (31ms)	100 (2.01h)	100 (7.00d)	
#9	程梓豪	400 (7.28d)	100 (665ms)	100 (5.43h)	100 (6.96d)	100 (2.16h)	
#10	隋钰涵	200 (369ms)	100 (337ms)	100 (32ms)			
#11	刘闯速	158 (2.00h)	48 (0ms)	10 (0ms)		100 (2.00h)	
#12	杨瑾硕	100 (7.01d)				100 (7.01d)	

## 作业

https://www.luogu.com.cn/contest/251033 (课上讲了 A ~ D 题, 课后作业是 E 题)

## 课堂表现

今天主要练了一些剪枝的题, 剪枝的题每道题得剪枝策略可能都不一样, 同学们关键是要抓住剪枝的核心, 针对每道题想它的剪枝策略。

## 课堂内容

B4033 [语言月赛 202409] 考试

先记录  $a[i] > b[i]$  的有  $\text{cnt1}$  个,  $a[i] == b[i]$  的有  $\text{cnt2}$  个,  $a[i] < b[i]$  的有  $\text{cnt3}$  个

1.  $\text{cnt1} > \text{cnt3}$  时, 直接输出 0
2.  $\text{cnt1} + \text{cnt2} > \text{cnt3}$  时, 直接输出  $\text{cnt3} + 1 - \text{cnt1}$
3. 把所有  $a[i] < b[i]$  的, 按照  $b[i] - a[i]$  的值从小到大排序, 然后从小往大处理

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

using namespace std;

const int maxn = 1000 + 5;
int a[maxn], b[maxn];

int main()
{
    int n; cin >> n;
    for (int i = 1; i <= n; ++i) cin >> a[i];
    for (int i = 1; i <= n; ++i) cin >> b[i];

    vector<int> vec;
    int cnt1 = 0, cnt2 = 0, cnt3 = 0;
    for (int i = 1; i <= n; ++i) {
        if (a[i] > b[i]) ++cnt1;
        else if (a[i] == b[i]) cnt2++;
        else vec.push_back(b[i] - a[i]), cnt3++;
    }

    if (cnt1 > cnt3) { cout << 0 << endl; return 0; }
    if (cnt1 + cnt2 > cnt3) { cout << cnt3 + 1 - cnt1 << endl; return 0; }

    int res = cnt2;
    cnt1 += cnt2;
    sort(vec.begin(), vec.end());
    for (int i : vec) {
        res += i;
        cnt3--;
        if (cnt1 > cnt3) break;
        res++;
        cnt1++;
        if (cnt1 > cnt3) break;
    }
    cout << res << endl;
    return 0;
}
```

## B3624 猫粮规划

```
// dfs 写法
#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;
}
```

```
// dp 写法
#include <bits/stdc++.h>

using namespace std;

const int maxn = 40 + 5;
int w[maxn];
int n, l, r;
int f[maxn][4005];

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

    f[0][0] = 1;
```

```

for (int i = 1; i <= n; ++i) {
    for (int j = 0; j <= 4000; ++j) {
        f[i][j] = f[i-1][j];
        if (j >= w[i]) f[i][j] += f[i-1][j-w[i]];
    }
}

int res = 0;
for (int i = 1; i <= r; ++i) res += f[n][i];
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;
}
```