剪枝

人员

陈奕然、王承周、司云心、褚锦轩、阮文璋 到课

上周作业检查

https://www.luogu.com.cn/contest/249994

2025-0601六队上课(综合混练)

报名 编辑比赛

题目数 5 | 报名人数 21

比赛说明 题目列表 排行榜

名次	参赛者	总分	A	B	C	D	E
#1	杨俊彦	500 (12.54h)	100 (2.19h)	100 (2.96h)	100 (2.40h)	100 (2.25h)	100 (2.74h)
#2	李锦澍	500 (1.15d)	100 (35ms)	100 (754ms)	100 (12.69h)	100 (1.96h)	100 (12.90h)
#3	徐思远	500 (4.41d)	100 (35ms)	100 (187ms)	100 (4.41d)	100 (696ms)	100 (257ms)
#4	阮文璋	500 (5.38d)	100 (39ms)	100 (1.64s)	100 (1.57d)	100 (4.88h)	100 (3.62d)
#5	袁晨峻	500 (12.76d)	100 (35ms)	100 (108ms)	100 (6.38d)	100 (353ms)	100 (6.38d)
#6	王承周	500 (15.02d)	100 (36ms)	100 (911ms)	100 (4.45d)	100 (4.44d)	100 (6.12d)
#7	褚锦轩	500 (31.74d)	100 (6.32d)	100 (6.34d)	100 (6.35d)	100 (6.36d)	100 (6.37d)
#8	周治润	400 (2.00h)	100 (35ms)	100 (246ms)	100 (2.00h)	100 (1.48s)	
#9	杨咏丞	400 (4.94d)	100 (1.21d)	100 (2.48d)	100 (1.25d)	100 (1.45s)	
#10	曹塬	300 (4.79d)	100 (36ms)	100 (3.55d)		100 (1.24d)	
#11	SSJ司云心	300 (14.58d)	10 (6.98h)	100 (2.48d)	90 (6.29d)	100 (5.52d)	
#12	龙沛轩	220 (2.00h)	100 (38ms)		20 (2.00h)	100 (1.57s)	
#13	李雨谦	200 (1.45s)	100 (37ms)			100 (1.42s)	
#14	王毅博	200 (1.65s)	100 (36ms)	100 (1.61s)			
#15	徐易	100 (6.36d)					100 (6.36d)

作业

https://www.luogu.com.cn/contest/251032 (课上讲了 A ~ D 题, 课后作业是 E 题, D 题不要求同学们得满分, 得 n^2 的暴力分即可)

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

U480698 Wandering

前缀和 + 前缀最大值 进行维护

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

using namespace std;

typedef long long LL;
const int maxn = 2e5 + 5;
int w[maxn];
LL p[maxn], pMax[maxn];

int main()
{
  int n; cin >> n;
  for (int i = 1; i <= n; ++i) cin >> w[i];
  for (int i = 1; i <= n; ++i) {
    p[i] = p[i-1] + w[i];
    pMax[i] = max(pMax[i-1], p[i]);
  }

  LL pos = 0, res = 0;
  for (int i = 1; i <= n; ++i) {
    res = max(res, pos + pMax[i]);
    pos += p[i];
  }
  cout << res << endl;
  return 0;
}
```

B3624 猫粮规划

```cpp
// 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 = l; 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;
}
```

U564690 异或和

这个题满分做法比较难一些, 同学们能掌握 O(n^2) 做法即可

O(n^2) 做法: 维护一个前缀异或值的数组, 这样就可以 O(1) 求区间异或值了

O(nlogn) 做法: 针对每个二进制位进行考虑

```
// O(n^2) 做法
#include <bits/stdc++.h>

using namespace std;

typedef long long LL;
const int maxn = 2e5 + 5;
int w[maxn], p[maxn];

int get_value(int l, int r) { return p[r] ^ p[l-1]; }

int main()
{
  int n; cin >> n;
  for (int i = 1; i <= n; ++i) cin >> w[i], p[i] = p[i-1] ^ w[i];

  LL res = 0;
  for (int i = 1; i <= n-1; ++i) {
    for (int j = i+1; j <= n; ++j)
      res += get_value(i, j);
  }
  cout << res << endl;
  return 0;
}
```

```
// O(nlogn) 做法
#include <bits/stdc++.h>

using namespace std;

typedef long long LL;
const int maxn = 2e5 + 5;
int w[maxn];
int n;

LL calc(int k) {
  int odd = 0, even = 1;
  LL sum = 0;
  for (int i = 1; i <= n; ++i) {
    sum ^= w[i];
    if ((sum>>k) % 2 == 1) ++odd;
    else ++even;
  }
  return (LL)odd * even;
}
```

```
int main()
{
  cin >> n;
  LL sum = 0;
  for (int i = 1; i <= n; ++i) cin >> w[i], sum += w[i];

  LL res = 0;
  for (int i = 0; i <= 30; ++i)
    res += calc(i)*(1<<i);
  cout << res-sum << endl;
  return 0;
}
```