

# 综合混练

## 人员

程梓豪、隋钰涵、赵熙羽、高健桓、杨瑾硕、董浩桢、牛同泽、秦显森、武敬哲、刘闯速、庞滢、于子珈 到课

## 上周作业检查

<https://www.luogu.com.cn/contest/240402>

2025-0406周日10:30

报名

编辑比赛

题目数

4

报名人数

14

比赛说明

题目列表

排行榜

名次	参赛者	总分	A	B	C	D
#1	程梓豪	400 (20.08h)	100 (4.87h)	100 (36ms)	100 (8.94h)	100 (6.28h)
#2	赵熙羽	400 (8.85d)	100 (76ms)	100 (31ms)	100 (2.39d)	100 (6.45d)
#3	隋钰涵	310 (6.55d)	100 (62ms)	100 (31ms)	10 (1.99h)	100 (6.46d)
#4	郭骐嘉	300 (6.46d)	100 (67ms)	100 (30ms)	100 (6.46d)	
#5	秦显森	300 (6.99d)	100 (66ms)	100 (31ms)	100 (6.99d)	
#6	高健桓	300 (18.61d)	100 (5.82d)	100 (5.82d)	100 (6.97d)	
#7	杨瑾硕	300 (20.86d)	100 (6.94d)	100 (6.94d)	100 (6.97d)	
#8	刘闯速	200 (98ms)	100 (67ms)	100 (31ms)		
#9	董浩桢	200 (2.03h)	100 (2.03h)	100 (31ms)		
#10	武敬哲	200 (3.08h)	100 (3.08h)	100 (30ms)	0	
#11	牛同泽	100 (66ms)	100 (66ms)			
#12	谢亚锴	100 (7.64h)		100 (7.64h)		

## 作业

<https://www.luogu.com.cn/contest/241022> (课上讲了 A ~ C 题, 课后作业是 D 题)

## 课堂表现

同学们课上做题表现不是特别好, 课下要再好好补补题。

今天存在有同学看题解的行为, 以后要进行改正, 不能再出现这种情况。

## 课堂内容

**B4303 [蓝桥杯青少年组省赛 2024] 字母移位**

考虑往左为正方向, 那么原问题本质上等价于:

- 第 1 个字符要变  $a_1 - a_2 + a_3 - a_4 + a_5 - a_6 + \dots$
- 第 2 个字符要变  $-a_2 + a_3 - a_4 + a_5 - a_6 + \dots$
- 第 3 个字符要变  $a_3 - a_4 + a_5 - a_6 + \dots$
- ...

所以, 可以维护一个数组, 数组的每一项分别为:  $a_1, -a_2, a_3, -a_4, a_5, -a_6, \dots$

第  $i$  个字符要进行的变化: 其实就是第  $i$  项一直到第  $n$  项的后缀和, 因此, 可以维护一个后缀和数组来  $O(1)$  求第  $i$  项字符的变化

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

using namespace std;

typedef long long LL;
const int maxn = 1e5 + 5;
char s[maxn];
int w[maxn], val[maxn];
LL suf[maxn];

char calc(char x, LL k) {
    k %= 26;
    if (k < 0) k += 26;

    x -= k;
    if (x < 'a') x += 26;
    return x;
}

int main()
{
    int n; cin >> n;
    cin >> (s+1);
    for (int i = 1; i <= n; ++i) {
        cin >> w[i];
        if (i&1) val[i] = w[i];
        else val[i] = -w[i];
    }

    for (int i = n; i >= 1; --i) suf[i] = suf[i+1] + val[i];

    for (int i = 1; i <= n; ++i) cout << calc(s[i], suf[i]);
    cout << endl;
    return 0;
}
```

比较直接的二分, 二分判断目前的 X 元能否买 mid 这个整数即可

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

using namespace std;

typedef long long LL;
LL a, b, c;

bool check(LL mid) {
    LL value = mid*a;
    int cnt = 0;
    while (mid) mid /= 10, cnt++;
    value += cnt*b;
    return value <= c;
}

int main()
{
    cin >> a >> b >> c;
    LL l = 1, r = 1000000000;
    while (l <= r) {
        LL mid = (l + r) / 2;
        if (check(mid)) l = mid+1;
        else r = mid-1;
    }
    cout << r << endl;
    return 0;
}
```

### U537157 souvenir

以每个点出发做一遍 bfs (一共 n 个点, 每次 bfs 都是  $O(n)$  的, 总时间复杂度:  $O(n^2)$ )

这个题在 bfs 模板的基础上, 多加了一个问题: 在希望最短路的情况下, 还希望 起点 到 目标点 的总价值尽可能大

所以在除了维护 dis 数组代表最短路的要求下, 还需要维护一个 f 数组, 代表起点到目标点的最大价值

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

using namespace std;

typedef long long LL;
const int maxn = 300 + 5;
int w[maxn];
int dis[maxn][maxn]; LL f[maxn][maxn];
char s[maxn][maxn];
int n;

void bfs(int x) {
```

```
queue<int> q; q.push(x); dis[x][x] = 0; f[x][x] = w[x];
while (!q.empty()) {
    int u = q.front(); q.pop();
    for (int i = 1; i <= n; ++i) {
        if (s[u][i]=='Y' && dis[x][i]==-1) {
            q.push(i); dis[x][i] = dis[x][u]+1; f[x][i] = f[x][u]+w[i];
        } else if (s[u][i]=='Y' && dis[x][u]+1==dis[x][i]) {
            f[x][i] = max(f[x][i], f[x][u]+w[i]);
        }
    }
}

int main()
{
    memset(dis, -1, sizeof(dis)); memset(f, 0, sizeof(f));

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

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

    int m; cin >> m;
    while (m -- ) {
        int u, v; cin >> u >> v;
        if (dis[u][v] == -1) cout << "Impossible" << endl;
        else cout << dis[u][v] << " " << f[u][v] << endl;
    }
    return 0;
}
```