

csp小学组2023题目讲解

人员

辛帅辰、方俊喆、刘祺、夏硕承、牛同泽、柳力玮、高健桓、谢亚锴、秦显森、赵熙羽 到课，齐振玮 线上

作业检查

赵熙羽 完成 2 道题

谢亚锴、夏硕承、柳立玮 完成 1 道题

其他同学 未完成

作业

<https://www.luogu.com.cn/contest/203581>, A B C 3道题目

课堂表现

大部分同学目前写代码能力明显有所下降，基本的暴力写起来都吃力了。

可以把之前做过的、但是并不熟练的题目再做一做，增加一下熟练度。

课堂内容

U478308 求合法对数

用桶记录 $b[c[i]]$ 中每个数出现的次数，之后就可以 $O(1)$ 查询 $a[i]$ 出现的次数了

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

using namespace std;

typedef long long LL;
const int maxn = 1e5 + 5;
int a[maxn], b[maxn], c[maxn];
int f[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];
    for (int i = 1; i <= n; ++i) cin >> c[i];

    for (int i = 1; i <= n; ++i) f[b[c[i]]]++;

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

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

U480698 Wandering

维护 前缀和数组 和 前缀最大值数组, 后面一层一层考虑

对于每一层的考虑, 可以 $O(1)$ 做

```
#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;
}
```

T504031 回文字符串 (palindrome)

左右 2 个变量, 从两边往中间遍历。

同时维护 2 个字符串, 随着遍历的过程维护字符串的改变。

当字符串相同或为回文时, 即可进行分割。

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

using namespace std;
```

```
bool check(string s, string e) {
    reverse(s.begin(), s.end());
    if (s == e) return true;
    return false;
}

int main() {
    string str; cin >> str;
    string s, e;
    int cnt = 0;
    for (int i = 0, j = (int)str.size()-1; i < j; i++, j--) {
        s += str[i], e += str[j];
        if (s==e || check(s,e)) {
            cnt += 2;
            s = "", e = "";
        }
    }

    int n = str.size();
    if (n%2 == 1) cnt++;
    else {
        if (s != "") cnt++;
    }

    if (cnt==0 || cnt==1) cout << "NO" << endl;
    else {
        cout << "YES" << endl;
        cout << cnt << endl;
    }
    return 0;
}
```

T504032 克隆机 (clone)

不需要一个一个考虑，只需要维护一个字母一次连续出现了多少个即可

```
// 暴力代码
#include <bits/stdc++.h>

using namespace std;

int main()
{
    int k; long long n;
    cin >> k >> n;
    for (int i = 1; ; i*=2) {
        for (char c = 'A'; c <= 'A'+k-1; c++) {
            if (i >= n) {
                cout << c << endl;
                return 0;
            } else n -= i;
        }
    }
}
```

```
    }  
}  
return 0;  
}
```

```
// 正确代码  
#include <bits/stdc++.h>  
  
using namespace std;  
  
int main()  
{  
    int k; long long n;  
    cin >> k >> n;  
    for (long long i = 1; ; i*=2) {  
        for (char c = 'A'; c <= 'A'+k-1; c++) {  
            if (i >= n) {  
                cout << c << endl;  
                return 0;  
            } else n -= i;  
        }  
    }  
    return 0;  
}
```