2021-11-14周赛(第一场)

T1

1. 题意

子序列含有"hello"

2. 思路

直接做就行

3. 参考代码

```cpp
#include <iostream>
#include <cstdio>
#include <algorithm>
#include <cstring>
#include <vector>

using namespace std;

string str;

void solve() {
    int cur = 0;
    for (char ch : str) {
        if (cur == 0 && ch == 'h') cur++;
        else if (cur == 1 && ch == 'e') cur++;
        else if (cur == 2 && ch == 'l') cur++;
        else if (cur == 3 && ch == 'l') cur++;
        else if (cur == 4 && ch == 'o') cur++;
    }

    cout << (cur == 5 ? "YES" : "NO") << endl;
}

int main() {
#ifdef LOCAL
    freopen("../in.txt", "r", stdin);
    freopen("../out.txt", "w", stdout);
#endif

    ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);
    cin >> str;
    solve();

    return 0;
}
```

T2

1. 题意

求最小哈密顿路径

2. 思路

状压dp

f(mask, i, j, k):

mask: mask里的每一位表示的是某个城市是否去过(1去过, 0没去过)

i: 表示最后到达的城市是i

j: 表示有无走过路(1有, 0无)

k: 表示有无开飞机(1有, 0无)

3. 参考代码

```
#include <iostream>
#include <cstdio>
#include <algorithm>
#include <cstring>
#include <vector>
#include <bitset>

using namespace std;

int n;
int g[22][22];

void solve() {
    int lim = 1 << n;
    int f[lim][n][2][2];
    memset(f, 0x3f, sizeof f);
    for (int i = 0; i < n; i++) {
        int ns = 1 << i;
        f[ns][i][0][0] = 0;
    }

    for (int mask = 0; mask < lim; mask++) {
        for (int i = 0; i < n; i++) {
            for (int j = 0; j <= 1; j++) {
                for (int k = 0; k <= 1; k++) {
                    for (int v = 0; v < n; v++) {
                        if (mask >> v & 1) continue;

                        int ns = mask + (1 << v);
                        // 普通
                        f[ns][v][j][k] = min(f[ns][v][j][k], f[mask][i][j][k] +
g[i][v]);
                        // 走路
                        if (j == 0) f[ns][v][1][k] = min(f[ns][v][1][k], f[mask]
[i][j][k]);
                        // 开飞机
                        if (k == 0) f[ns][v][j][1] = min(f[ns][v][j][1], f[mask]
[i][j][k] + 3 * g[i][v]);
```

```
                    }
                }
            }
        }
    }

    int ans = 1e9;
    for (int i = 0; i < n; i++) {
        ans = min(ans, f[lim - 1][i][1][1]);
    }
    cout << ans << endl;
}

int main() {
#ifdef LOCAL
    freopen("../in.txt", "r", stdin);
    freopen("../out.txt", "w", stdout);
#endif

    ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);
    cin >> n;
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            cin >> g[i][j];
        }
    }
    solve();

    return 0;
}
```

T3

1. 题意

非递减字符串方案数

2. 思路

可以推出递推方程:

f(i, 'a') = f(i - 1, 'a')

f(i, 'b') = f(i - 1, 'a') + f(i - 1, 'b')

f(i, 'c') = f(i - 1, 'a') + f(i - 1, 'b') + f(i - 1, 'c')

...

由于范围过大, 需要使用矩阵快速幂进行优化

3. 参考代码

```
#include <iostream>
#include <cstdio>
#include <algorithm>
#include <cstring>
#include <vector>

using namespace std;
```

```
#define ll long long

const int N = 33;
const ll MOD = 20211121;

struct matrix {
    int r, c;
    ll s[N][N];

    matrix(int r = 0, int c = 0) : r(r), c(c) {
        memset(s, 0, sizeof s);
    }

    matrix operator*(const matrix &that) const {
        matrix res = matrix(r, that.c);
        for (int i = 1; i <= res.r; i++) {
            for (int j = 1; j <= res.c; j++) {
                for (int k = 1; k <= c; k++) {
                    res.s[i][j] = (res.s[i][j] + s[i][k] * that.s[k][j] % MOD) %
MOD;
                }
            }
        }
        return res;
    }
};

ll n;
matrix m1, m2;

// 默认 b >= 1
matrix qsm(matrix a, ll b) {
    matrix res = a;
    b--;
    while (b) {
        if (b & 1) res = res * a;
        a = a * a;
        b >>= 1;
    }

    return res;
}

void init() {
    m1 = matrix(26, 26);
    m2 = matrix(26, 1);

    for (int i = 1; i <= 26; i++) {
        for (int j = 1; j <= i; j++) {
            m1.s[i][j] = 1;
        }
    }

    for (int i = 1; i <= 26; i++) m2.s[i][1] = 1;
}

void print(matrix &mat) {
    cout << mat.r << " " << mat.c << endl;

    for (int i = 1; i <= mat.r; i++) {
        for (int j = 1; j <= mat.c; j++) {
            cout << mat.s[i][j] << " ";
        }
        cout << endl;
    }
    cout << endl;
}

void solve() {
    init();

    if (n == 1){
        cout << 26 << endl;
        return;
    }

    m1 = qsm(m1, n - 1);
    matrix res = m1 * m2;
    ll ans = 0;
    for (int i = 1; i <= 26; i++) {
        ans = (ans + res.s[i][1]) % MOD;
    }
    cout << ans << endl;
}

int main() {
#ifdef LOCAL
    freopen("../in.txt", "r", stdin);
    freopen("../out.txt", "w", stdout);
#endif

    ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);
    cin >> n;
    solve();

    return 0;
}
```

T4

有手就行(不是)