Олимпиада СПбГУ по информатике 2018/19 учебного года

Васильев Даниил Алексеевич

A	B	C	D	E	F	Sum
100	100	100	60	100	24	484

Task A (100)

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

using namespace std;

void run() {
    int n, m;
    cin >> n >> m;
    while ( n < m)
        n *= 2;
    if (n == m)
        cout << "Yes";
    else
        cout << "No";
}

signed main() { ios::sync_with_stdio(0); cin.tie(0); run(); }
```

Task B (100)

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

using namespace std;

void run() {
    int n;
    cin >> n;
    string s;
    cin >> s;
    for (int i = 1; i < n; i++) {
        if (s[i - 1] == 'o' && s[i] == 'r') {
            cout << "Yes";
            return;
        }
        swap(s[i - 1], s[i]);
        if ((s[i - 1] == 'o' && s[i] == 'r') || (i > 1 && s[i - 2] == 'o' && s[i - 1] == 'r') || (
            i + 1 < n && s[i] == 'o' && s[i + 1] == 'r')) {
            cout << "Yes";
            return;
        }
        swap(s[i - 1], s[i]);
    }
    cout << "No";
}

signed main() { ios::sync_with_stdio(0); cin.tie(0); run(); }
```

Task C (100)

```cpp
#include <bits/stdc++.h>
using namespace std;
#define int long long
#define rep(a, l, r) for (int a = l; a < r; a++)
#define pb push_back
#define vint vector<int>


const int N = 1e5 + 1;
vint t[N];
int siz[N];
int ans[N];
void calcSiz(int cur, int p = -1) {
    siz[cur] = 1;
    for (int i : t[cur])
        if (i != p) {
            calcSiz(i, cur);
            siz[cur] += siz[i];
        }
}

int n;
void findAns(int cur, int p = -1) {
    ans[cur] = n - siz[cur] + 1;
    for (int i : t[cur])
        if (i != p) {
            ans[cur] = max(ans[cur], siz[i] + 1);
            findAns(i, cur);
        }
}

void run() {

    cin >> n;
    rep(i, 1, n) {
        int v, u;
        cin >> v >> u;
        t[v].pb(u);
        t[u].pb(v);
    }

    memset(siz, 0, sizeof siz);
    memset(ans, 0, sizeof ans);
    calcSiz(1);
    findAns(1);
    for (int i = 1; i <= n; i++)
        cout << ans[i] << " ";
}

signed main() { ios::sync_with_stdio(0); cin.tie(0); run(); }
```

Task D (60)

```cpp
#include <bits/stdc++.h>
using namespace std;
#define int long long
#define rep(a, l, r) for (int a = l; a < r; a++)
#define pb push_back
#define vint vector<int>

string todo;
int t, n, p;


/*
012345
345678
678012
*/
/*
012345
234567
456780
678012
801234
*/
string s;
void split12(int T) {
    for (int i = 0; i < n; i++) {
        cout << char('a' + i * T);

        for (int j = 0; j < p - 1; j++)
            cout << s[(i * T + j) % 9];
        cout << " ";
    }
    cout << endl;
}

string nw[4];
void merge12() {
    string ss = "_____";
    for (int i = 0; i <= n / 2; i++) {
        int pos = nw[i][0] - 'a';
        for (int j = 1; j < p; j++)
            ss[(pos + j - 1) % 9] = nw[i][j];
    }
    cout << ss << endl;
}

void split3() {}
void merge3() {}

void run() {
    cin >> todo;
    cin >> t >> n >> p;

    for (; t; t--) {
        if (todo == "split") {
            cin >> s;
            if (n <= 3)
                split12(3);
            else if (n == 5)
                split12(2);
            else
                split3();
        } else {
            for (int i = 0; i <= n / 2; i++)
                cin >> nw[i];
            if (n <= 5)
                merge12();
            else
                merge3();
```

```
        }
    }
}

signed main() { ios::sync_with_stdio(0); cin.tie(0); run(); }
```

Task E (100)

```python
def dist(a, b):
    return (a[0] - b[0]) * (a[0] - b[0]) + (a[1] - b[1]) * (a[1] - b[1])


n = int(input())
c = []
mn = 0
for i in range(n):
    x, y = map(int, input().split())
    mn = min(mn, x, y)
    c.append((x, y))
px, py = map(int, input().split())
qx, qy = map(int, input().split())
dx, dy = qx - px, qy - py
mul = 1000000000000000000000000000000
dx *= mul
dy *= mul
qx, qy = px + dx, py + dy
mn = min(mn, px, py, qx, qy)
px, py, qx, qy = px - mn, py - mn, qx - mn, qy - mn
q = (qx, qy)
ans, p = -1, -1

for i in range(n):
    c[i] = (c[i][0] - mn, c[i][1] - mn)
    if ans == -1 or dist(q, c[i]) < ans:
        ans = dist(q, c[i])
        p = i + 1
    elif dist(q, c[i]) == ans:
        p = -1
print(p)
```

Task F (24)

```cpp
#include <bits/stdc++.h>
using namespace std;
#define int long long
#define rep(a, l, r) for (int a = l; a < r; a++)
#define pb push_back
#define vint vector<int>

const int N = 30, K = 30;
const long long inf = 1000000000000000000ll;
long long a[N][2];
long long n, k;

long long dp[N][N * K + 1][N * K + 1];
int get(int i, int j, int l) {
    if (i >= 0 && i < n  && j >= 0 && j <= n * k && l >= 0 && l <= n * k)
        return dp[i][j][l];
    return -inf;
}

void run() {
    cin >> n >> k;

    rep(i, 0, n)
        cin >> a[i][0] >> a[i][1];

    rep(i, 0, n)
        rep(j, 0, n * k + 1)
            rep(l, 0, n * k + 1)
                dp[i][j][l] = -inf;
    dp[0][max(0ll, k - a[0][0])][0] = max(dp[0][max(0ll, k - a[0][0])][0], min(k, a[0][0]));
    dp[0][0][max(0ll, k - a[0][1])] = max(dp[0][0][max(0ll, k - a[0][1])], min(k, a[0][1]));

    rep(i, 1, n) {
        //j = 0, l = 0
        for (int z = 0; z + k <= a[i][0]; z++)
            for (int x = 0; x <= a[i][1]; x++)
                dp[i][0][0] = max(dp[i][0][0], dp[i - 1][z][x] + z + x + k);

        for (int z = 0; z <= a[i][0]; z++)
            for (int x = 0; x + k <= a[i][1]; x++)
                dp[i][0][0] = max(dp[i][0][0], dp[i - 1][z][x] + z + x + k);

        //j = 0
        rep(l, 1, n * k + 1) {
            for (int z = 0; z + k <= a[i][0]; z++)
                dp[i][0][l] = max(dp[i][0][l], get(i - 1, z, l + a[i][1]) + z + k + a[i][1]);

            for (int z = 0; z <= a[i][0]; z++)
                dp[i][0][l] = max(dp[i][0][l], get(i - 1, z, l + a[i][1] - k) + z + a[i][1]);
        }

        //l = 0
        rep(j, 1, n * k + 1) {
            for (int z = 0; z + k <= a[i][1]; z++)
                dp[i][j][0] = max(dp[i][j][0], get(i - 1, j + a[i][0], z) + z + k + a[i][0]);

            for (int z = 0; z <= a[i][1]; z++)
                dp[i][j][0] = max(dp[i][j][0], get(i - 1, j + a[i][0] - k, z) + z + a[i][0]);
        }

        rep(j, 1, n * k + 1)
            rep(l, 1, n * k + 1) {
                dp[i][j][l] = max(dp[i][j][l], get(i - 1, j + a[i][0], l + a[i][1] - k) + a[i][0]
                    + a[i][1]);
                dp[i][j][l] = max(dp[i][j][l], get(i - 1, j + a[i][0] - k, l + a[i][1]) + a[i][0]
                    + a[i][1]);
            }
    }
}
```

```
        long long ans = 0;
        rep(j, 0, n * k)
            rep(l, 0, n * k)
                ans = max(ans, dp[n - 1][j][l]);
        cout << ans;
}

/*
ab -> a^b

*/
signed main() { ios::sync_with_stdio(0); cin.tie(0); run(); }
```