

Олимпиада «Ломоносов» по информатике
2023-2024 учебный год. Заключительный тур
Работа участника с id заявки 1234918, логином inf24f_102

Сводный итог по всем задачам в проверяющей системе

Run ID	Time	User name	Problem	Language	Result	Tests	Score
6	0:07:24	inf24f_102	1	python3	OK	28	100
29	0:33:55	inf24f_102	2	python3	OK	28	100
54	1:01:38	inf24f_102	3	python3	OK	28	100
148	2:17:41	inf24f_102	4	g++	Partial solution	3	10
200	2:53:24	inf24f_102	5	g++	OK	22	100
410 технических баллов							
82 итоговых балла							

Посылка по задаче 1

```
[1] from functools import lru_cache
[2]
[3] @lru_cache(maxsize=None, typed=False)
[4] def T(i):
[5]     if i <= 1:
[6]         return 0
[7]     if i == 2:
[8]         return 1
[9]     return T(i - 3) + T(i - 2) + T(i - 1)
[10]
[11] tri = [T(i) for i in range(3, 101) if T(i) <= 2 ** 24][::-1]
[12]
[13] def cnt(x):
[14]     res = 0
[15]     for v in tri:
[16]         if x >= v:
[17]             x -= v
[18]             res += 1
[19]     return res
[20]
[21] n = int(input())
[22] A = [int(input()) for _ in range(n)]
[23] print(sum(cnt(x) % 2 == 1 for x in A))
```

Посылка по задаче 2

```
[1] from fractions import Fraction
[2] pos = 0
[3] cnt = {"W": [], "R": [], "O": [], "Y": [], "G": [], "C": [], "B": [], "V": [], "D": []}
[4]
[5] def solve(depth=0):
[6]     global pos
[7]     c = s[pos]
[8]     pos += 1
[9]     if c == "Q":
[10]         for _ in range(4):
[11]             solve(depth + 1)
[12]     else:
[13]         cnt[c].append(depth)
[14]
[15] s = input()
[16] solve()
[17] mx = max(max(x, default=0) for x in cnt.values())
[18]
[19] total = 4 ** mx
[20] add = [Fraction(4 ** (mx - i), total) for i in range(mx + 1)]
[21] result = {"W": 0, "R": 0, "O": 0, "Y": 0, "G": 0, "C": 0, "B": 0, "V": 0, "D": 0}
[22] for k, v in cnt.items():
[23]     for x in v:
[24]         result[k] += add[x]
[25] answer = max(result, key=lambda x: (result[x], x))
[26] print(answer)
[27] if result[answer] == 1:
[28]     print("1.0")
[29] else:
[30]     print("0.", end="")
[31]     i = 1
[32]     while result[answer]:
[33]         if Fraction(1, 2 ** i) <= result[answer]:
[34]             print(end="1")
[35]             result[answer] -= Fraction(1, 2 ** i)
[36]         else:
[37]             print(end="0")
[38]         i += 1
```

Посылка по задаче 3

```
[1] d = {">": 1, ">!": 2, ">!!": 3, ">!!!": 4, ">?": 5, "<": 6, "<!": 7, "<!!": 8, "<!!!": 9, "<?": 10}
[2] rd = {v: k for k, v in d.items()}
[3]
[4] def translate(s):
[5]     if s == "()":
[6]         return 0
[7]     res = 0
[8]     M = []
[9]     while s:
[10]         cur = ""
[11]         while cur not in d:
[12]             cur = s[-1] + cur
[13]             s = s[:-1]
[14]         M.append(d[cur])
[15]     M.reverse()
[16]     mul = 1
[17]     for x in M:
[18]         res += mul * x
[19]         mul *= 10
[20]     return res
[21]
[22] def make(x):
[23]     if x == 0:
[24]         return "("
[25]     result = ""
[26]     mul = 1
[27]     while x:
[28]         v = x % (mul * 10) // mul
[29]         if v == 0:
[30]             v = 10
[31]         x -= v * mul
[32]         result += rd[v]
[33]         mul *= 10
[34]     return result
[35]
[36] n = int(input())
[37] A = [input() for _ in range(n)]
[38] val = [translate(x) for x in A]
[39]
[40] best_diff = -1
[41] best_sum = 10000000
[42] ans_k = ans_l = -1
[43]
[44] for k in range(n):
[45]     for l in range(k + 1, n):
[46]         if abs(val[k] - val[l]) > best_diff or (abs(val[k] -
[47] val[l]) == best_diff and l + k < best_sum):
[48]             best_diff = abs(val[k] - val[l])
[49]             best_sum = l + k
[50]             ans_k = k + 1
[51]             ans_l = l + 1
[52] print(make(ans_k))
    print(make(ans_l))
```

Посылка по задаче 4

```
[1] #include <bits/stdc++.h>
[2] #pragma GCC optimize("O3")
[3] using namespace std;
[4]
[5] int32_t main() {
[6]     ios::sync_with_stdio(false);
[7]     cin.tie(0);
[8]
[9]     int n, m, r1, r2, r3;
[10]    cin >> n >> m >> r1 >> r2 >> r3;
[11]    vector<vector<int>> Acpy(n, vector<int>(m));
[12]    for (int i = 0; i < n; i++) {
[13]        for (int j = 0; j < m; j++) {
[14]            cin >> Acpy[i][j];
[15]        }
[16]    }
[17]    vector<int> order = {r1, r2, r3};
[18]    sort(order.begin(), order.end());
[19]    int ans = 0;
[20]    do {
[21]        int result = 0;
[22]        vector<vector<int>> A = Acpy;
[23]
[24]        {
[25]            vector<vector<int>> dist(n, vector<int>(m, 0));
[26]            vector<vector<pair<int, int>>> pr(n, vector<pair<int, int>>(m, {-1, -1}));
[27]            set<pair<int, pair<int, int>>, greater<>> q;
[28]            dist[order[0]][0] = A[order[0]][0];
[29]            q.insert({dist[order[0]][0], {order[0], 0}});
[30]            while (!q.empty()) {
[31]                auto[i, j] = q.begin()->second;
[32]                q.erase(q.begin());
[33]                if (j == m - 1) {
[34]                    continue;
[35]                }
[36]                if (i - 1 >= 0 && dist[i][j] + A[i - 1][j + 1] > dist[i - 1][j + 1]) {
[37]                    q.erase({dist[i - 1][j + 1], {i - 1, j + 1}});
[38]                    dist[i - 1][j + 1] = dist[i][j] + A[i - 1][j + 1];
[39]                    pr[i - 1][j + 1] = {i, j};
[40]                    q.insert({dist[i - 1][j + 1], {i - 1, j + 1}});
[41]                }
[42]                if (dist[i][j] + A[i][j + 1] > dist[i][j + 1]) {
[43]                    q.erase({dist[i][j + 1], {i, j + 1}});
[44]                    dist[i][j + 1] = dist[i][j] + A[i][j + 1];
[45]                    pr[i][j + 1] = {i, j};
[46]                    q.insert({dist[i][j + 1], {i, j + 1}});
[47]                }
[48]                if (i + 1 < n && dist[i][j] + A[i + 1][j + 1] > dist[i + 1][j + 1]) {
[49]                    q.erase({dist[i + 1][j + 1], {i + 1, j + 1}});
[50]                    dist[i + 1][j + 1] = dist[i][j] + A[i + 1][j + 1];
[51]                    pr[i + 1][j + 1] = {i, j};
[52]                    q.insert({dist[i + 1][j + 1], {i + 1, j + 1}});
[53]                }
[54]            }
[55]
[56]            int mx = 0;
[57]            for (int i = 1; i < n; i++) {
[58]                if (dist[i][m - 1] > dist[mx][m - 1]) {
[59]                    mx = i;
[60]                }
[61]            }
[62]            result += dist[mx][m - 1];
[63]            int x = mx, y = m - 1;
[64]            while (make_pair(x, y) != make_pair(-1, -1)) {
[65]                A[x][y] = 0;
[66]                tie(x, y) = pr[x][y];
[67]            }
[68]        }
[69]
[70]        vector<vector<int>> dist(n, vector<int>(m, 0));
[71]        vector<vector<pair<int, int>>> pr(n, vector<pair<int, int>>(m, {-1, -1}));
[72]        set<pair<int, pair<int, int>>, greater<>> q;
[73]        dist[order[1]][0] = A[order[1]][0];
[74]        q.insert({dist[order[1]][0], {order[1], 0}});
[75]        while (!q.empty()) {
[76]            auto[i, j] = q.begin()->second;
[77]            q.erase(q.begin());
[78]            if (j == m - 1) {
[79]                continue;
[80]            }
[81]            if (i - 1 >= 0 && dist[i][j] + A[i - 1][j + 1] > dist[i - 1][j + 1]) {
[82]                q.erase({dist[i - 1][j + 1], {i - 1, j + 1}});
[83]                dist[i - 1][j + 1] = dist[i][j] + A[i - 1][j + 1];
[84]                pr[i - 1][j + 1] = {i, j};
[85]                q.insert({dist[i - 1][j + 1], {i - 1, j + 1}});
[86]            }
[87]            if (dist[i][j] + A[i][j + 1] > dist[i][j + 1]) {
[88]                q.erase({dist[i][j + 1], {i, j + 1}});
[89]                dist[i][j + 1] = dist[i][j] + A[i][j + 1];
[90]                pr[i][j + 1] = {i, j};
[91]                q.insert({dist[i][j + 1], {i, j + 1}});
[92]            }
[93]            if (i + 1 < n && dist[i][j] + A[i + 1][j + 1] > dist[i + 1][j + 1]) {
```

```

[94]         q.erase({dist[i + 1][j + 1], {i + 1, j + 1}});
[95]         dist[i + 1][j + 1] = dist[i][j] + A[i + 1][j + 1];
[96]         pr[i + 1][j + 1] = {i, j};
[97]         q.insert({dist[i + 1][j + 1], {i + 1, j + 1}});
[98]     }
[99] }
[100]
[101] int mx = 0;
[102] for (int i = 1; i < n; i++) {
[103]     if (dist[i][m - 1] > dist[mx][m - 1]) {
[104]         mx = i;
[105]     }
[106] }
[107] result += dist[mx][m - 1];
[108] int x = mx, y = m - 1;
[109] while (make_pair(x, y) != make_pair(-1, -1)) {
[110]     A[x][y] = 0;
[111]     tie(x, y) = pr[x][y];
[112] }
[113] }
[114] {
[115]     vector<vector<int>> dist(n, vector<int>(m, 0));
[116]     vector<vector<pair<int, int>>> pr(n, vector<pair<int, int>>(m, {-1, -1}));
[117]     set<pair<int, pair<int, int>>, greater<>> q;
[118]     dist[order[2]][0] = A[order[2]][0];
[119]     q.insert({dist[order[2]][0], {order[2], 0}});
[120]     while (!q.empty()) {
[121]         auto[i, j] = q.begin()->second;
[122]         q.erase(q.begin());
[123]         if (j == m - 1) {
[124]             continue;
[125]         }
[126]         if (i - 1 >= 0 && dist[i][j] + A[i - 1][j + 1] > dist[i - 1][j + 1]) {
[127]             q.erase({dist[i - 1][j + 1], {i - 1, j + 1}});
[128]             dist[i - 1][j + 1] = dist[i][j] + A[i - 1][j + 1];
[129]             pr[i - 1][j + 1] = {i, j};
[130]             q.insert({dist[i - 1][j + 1], {i - 1, j + 1}});
[131]         }
[132]         if (dist[i][j] + A[i][j + 1] > dist[i][j + 1]) {
[133]             q.erase({dist[i][j + 1], {i, j + 1}});
[134]             dist[i][j + 1] = dist[i][j] + A[i][j + 1];
[135]             pr[i][j + 1] = {i, j};
[136]             q.insert({dist[i][j + 1], {i, j + 1}});
[137]         }
[138]         if (i + 1 < n && dist[i][j] + A[i + 1][j + 1] > dist[i + 1][j + 1]) {
[139]             q.erase({dist[i + 1][j + 1], {i + 1, j + 1}});
[140]             dist[i + 1][j + 1] = dist[i][j] + A[i + 1][j + 1];
[141]             pr[i + 1][j + 1] = {i, j};
[142]             q.insert({dist[i + 1][j + 1], {i + 1, j + 1}});
[143]         }
[144]     }
[145] }
[146] int mx = 0;
[147] for (int i = 1; i < n; i++) {
[148]     if (dist[i][m - 1] > dist[mx][m - 1]) {
[149]         mx = i;
[150]     }
[151] }
[152] result += dist[mx][m - 1];
[153] int x = mx, y = m - 1;
[154] while (make_pair(x, y) != make_pair(-1, -1)) {
[155]     A[x][y] = 0;
[156]     tie(x, y) = pr[x][y];
[157] }
[158] }
[159]
[160] ans = max(ans, result);
[161] } while (next_permutation(order.begin(), order.end()));
[162]
[163] cout << ans;
[164] return 0;
[165] }

```

Посылка по задаче 5

```
[1] #include <bits/stdc++.h>
[2] #pragma GCC optimize("O3")
[3] using namespace std;
[4]
[5]
[6] int32_t main() {
[7]     ios::sync_with_stdio(false);
[8]     cin.tie(0);
[9]
[10]     string s, t;
[11]     cin >> s >> t;
[12]     vector<int> cnt(94), cur_cnt(94);
[13]     for (char x : t) {
[14]         cnt[x - 33]++;
[15]     }
[16]     int need = 0;
[17]     for (int x: cnt) {
[18]         if (x) need++;
[19]     }
[20]
[21]
[22]     int la = 0, ra = 0;
[23]
[24]     int have = 0, j = -1;
[25]     while (j + 1 < s.size() && have != need) {
[26]         j++;
[27]         cur_cnt[s[j] - 33]++;
[28]         if (cur_cnt[s[j] - 33] == cnt[s[j] - 33]) {
[29]             have++;
[30]         }
[31]     }
[32]     if (have == need) {
[33]         if (ra - la == 0 || j + 1 - 0 < ra - la) {
[34]             la = 0;
[35]             ra = j + 1;
[36]         }
[37]     }
[38]     for (int i = 0; i <= s.size(); i++) {
[39]         if (cur_cnt[s[i] - 33] == cnt[s[i] - 33]) have--;
[40]         cur_cnt[s[i] - 33]--;
[41]
[42]         while (j + 1 < s.size() && have != need) {
[43]             j++;
[44]             cur_cnt[s[j] - 33]++;
[45]             if (cur_cnt[s[j] - 33] == cnt[s[j] - 33]) {
[46]                 have++;
[47]             }
[48]         }
[49]         if (have == need) {
[50]             if (ra - la == 0 || j + 1 - (i + 1) < ra - la) {
[51]                 la = i + 1;
[52]                 ra = j + 1;
[53]             }
[54]         }
[55]     }
[56]     cout << string(s.begin() + la, s.begin() + ra);
[57]     return 0;
[58] }
```