Олимпиада «Ломоносов» по информатике
2023-2024 учебный год. Заключительный тур
Работа участника с id заявки 1198744, логином inf24f_250

Сводный итог по всем задачам в проверяющей системе

Run ID Time User name Problem Language Result Tests Score
458 3:06:02 inf24f_250 1 python3 OK 28 100
54 0:27:48 inf24f_250 2 python3 OK 28 100
205 1:20:03 inf24f_250 3 python3 OK 28 100
435 2:53:53 inf24f_250 4 g++ Partial solution 6 25
605 3:46:11 inf24f_250 5 g++ OK 22 100
631 3:51:53 inf24f_250 6 python3 Partial solution 3 20
445 технических баллов
74 итоговых балла

Посылка по задаче 1

```
[1] tetra = [0, 0, 0, 1]
[2]
[3] for i in range(27):
[4]     tetra.append(tetra[-1] + tetra[-2] + tetra[-3] + tetra[-4])
[5]
[6] tetra = tetra[4:]
[7] tetra.reverse()
[8]
[9] def one_cnt(i):
[10]    ones = 0
[11]    for tet in tetra:
[12]        if tet <= i:
[13]            ones += 1
[14]            i -= tet
[15]    return ones
[16]
[17] n = int(input())
[18] ans = 0
[19] for i in range(n):
[20]    a = int(input())
[21]    cnt = one_cnt(a)
[22]    if cnt % 2 == 0:
[23]        ans += 1
[24] print(ans)
```

Посылка по задаче 2

```
[1] from fractions import Fraction
[2]
[3] tree = input()
[4]
[5] colors = {}
[6]
[7] def parse(at: int, lv: Fraction) -> int:
[8]     # print(at, lv)
[9]     orig = at
[10]    if tree[at] != "Q":
[11]        clr = tree[at]
[12]        colors[clr] = colors.get(clr, 0) + lv
[13]        return 1
[14]    at += 1
[15]    for i in range(8):
[16]        at += parse(at, lv / 8)
[17]    # print(tree[orig:at])
[18]    return at - orig
[19]
[20] parse(0, Fraction(1))
[21]
[22] ansc = ""
[23] ansv = -1
[24] for clr in colors:
[25]    if colors[clr] > ansv or (colors[clr] == ansv and clr < ansc):
[26]        ansc = clr
[27]        ansv = colors[clr]
[28]
[29] print(ansc)
[30]
[31] if ansv == 1:
[32]    print(1.0)
[33]    exit(0)
[34] elif ansv == 0:
[35]    print(0.0)
[36]    exit(0)
[37]
[38] print("0.", end="")
[39] base = Fraction(1, 2)
[40] while ansv:
[41]    if base <= ansv:
[42]        ansv -= base
[43]        print("1", end="")
[44]    else:
[45]        print("0", end="")
[46]    base /= 2
```

Посылка по задаче 3

```
parseorder = [
    "i((", "I((", "J))", "j))", "i(", "I(", "J)", "j)", "i", "I", "J", "j"
]

vals = {
    "i": 1,
    "i(": 2,
    "i((": 3,
    "I": 4,
    "I(": 5,
    "I((": 6,
    "J": 7,
    "J)": 8,
    "J))": 9,
    "j": 10,
    "j)": 11,
    "j))": 12
}

inv_vals = {}
for key in vals:
    inv_vals[vals[key]] = key

def fromelf(s):
    if s == "[]":
        return 0
    res = 0
    pw = 1
    while s:
        adv = 0
        for num in parseorder:
            if s.startswith(num):
                res += vals[num] * pw
                adv = len(num)
                break
        pw *= 12
        s = s[adv:]
    return res

def toelf(i):
    dec = 0
    res = []
    if i == 0:
        return "[]"
    while i:
        digit = i % 12
        if digit == 0:
            digit = 12
            dec = 1
        # print(i, digit)
        # if digit <= 0:
        #     digit += 12
        #     dec = 1
        # else:
        #     dec = 0
        res.append(inv_vals[digit])
        i //= 12
        if dec:
            dec = 0
            i -= 1
    return "".join(res)

n = int(input())

chests = [fromelf(input()) for i in range(n)]

ansk = -1
ansl = -1
mxans = -1
for k in range(n):
    for l in range(k + 1, n):
        ans_at = abs(chests[k] - chests[l])
        if ans_at > mxans or (ans_at == mxans and k + l > ansk + ansl):
            ansk = k
            ansl = l
            mxans = ans_at
print(toelf(ansk + 1))
print(toelf(ansl + 1))
```

Посылка по задаче 4

```
[1] #include <bits/stdc++.h>
[2]
[3] using namespace std;
[4]
[5] #define int long long
[6] #define uint unsigned long long
[7] #define vec vector
[8] #define all(a) a.begin(), a.end()
[9] #define rall(a) a.rbegin(), a.rend()
[10]#define umap unordered_map
[11]#define uset unordered_set
[12]
[13]int r, c;
[14]vec<vec<int>> table;
[15]vec<vec<vec<vec<int>>>> dp;
[16]
[17]int shrooms(int b1, int b2, int b3, int col) {
[18]    set<int> pos{b1, b2, b3};
[19]    int res = 0;
[20]    for (auto p : pos) {
[21]        res += table[p][col];
[22]    }
[23]    return res;
[24]}
[25]
[26]bool valid(int pos) { return pos >= 0 && pos < r; }
[27]
[28]void calc(int i, int j, int k, int col) {
[29]    for (int di = -1; di <= 1; ++di) {
[30]        for (int dj = -1; dj <= 1; ++dj) {
[31]            for (int dk = -1; dk <= 1; ++dk) {
[32]                int oi = i + di;
[33]                int oj = j + dj;
[34]                int ok = k + dk;
[35]                if (valid(oi) && valid(oj) && valid(ok)) {
[36]                    dp[i][j][k][col] =
[37]                        max(dp[i][j][k][col],
[38]                            dp[oi][oj][ok][col - 1] + shrooms(i, j, k, col));
[39]                }
[40]            }
[41]        }
[42]    }
[43]}
[44]
[45]signed main() {
[46]    ios_base::sync_with_stdio(0);
[47]    cin.tie(0);
[48]
[49]    cin >> r >> c;
[50]
[51]    dp.resize(r, vec(r, vec(r, vec(c, 0ll))));
[52]
[53]    table.resize(r, vec(c, 0ll));
[54]
[55]    int b1, b2, b3;
[56]    cin >> b1 >> b2 >> b3;
[57]
[58]    for (auto& row : table) {
[59]        for (auto& cell : row) {
[60]            cin >> cell;
[61]        }
[62]    }
[63]
[64]    dp[b1][b2][b3][0] += shrooms(b1, b2, b3, 0);
[65]
[66]    for (int col = 1; col < c; ++col) {
[67]        for (int i = 0; i < r; ++i) {
[68]            for (int j = 0; j < r; ++j) {
[69]                for (int k = 0; k < r; ++k) {
[70]                    calc(i, j, k, col);
[71]                }
[72]            }
[73]        }
[74]    }
[75]
[76]    int ans = -1;
[77]    for (int i = 0; i < r; ++i) {
[78]        for (int j = 0; j < r; ++j) {
[79]            for (int k = 0; k < r; ++k) {
[80]                int ans_at = dp[i][j][k][c - 1];
[81]                if (ans_at > ans) {
[82]                    ans = ans_at;
[83]                }
[84]            }
[85]        }
[86]    }
[87]
[88]    cout << ans << endl;
[89]}
```

Посылка по задаче 5

```
[1]  #include <bits/stdc++.h>
[2]
[3]  using namespace std;
[4]
[5]  #define int long long
[6]  #define uint unsigned long long
[7]  #define vec vector
[8]  #define all(a) a.begin(), a.end()
[9]  #define rall(a) a.rbegin(), a.rend()
[10] #define umap unordered_map
[11] #define uset unordered_set
[12]
[13] vec<int> sub_freqmaps(vec<int>& big, vec<int>& small) {
[14]     vec<int> res(127);
[15]     for (int i = 0; i < 127; ++i) {
[16]         res[i] = big[i] - small[i];
[17]     }
[18]     return res;
[19] }
[20]
[21] bool compat(vec<int>& want, vec<int>& have) {
[22]     bool res = true;
[23]     for (int i = 0; i < 127; ++i) {
[24]         res = res && (want[i] <= have[i]);
[25]     }
[26]     return res;
[27] }
[28]
[29] int n;
[30] vec<int> want_freqmap(127);
[31] vec<vec<int>> orpref_freqs;
[32] vec<int> alan(127);
[33]
[34] int search(int from) {
[35]     int l = 0;
[36]     int r = n - from + 1;
[37]     bool found = false;
[38]     while (l < r - 1) {
[39]         int mid = (l + r) / 2;
[40]         int to = mid + from - 1;
[41]         auto have_freqmap = sub_freqmaps(
[42]             orpref_freqs[to], from == 0 ? alan : orpref_freqs[from - 1]);
[43]         if (compat(want_freqmap, have_freqmap)) {
[44]             found = true;
[45]             r = mid;
[46]         } else {
[47]             l = mid;
[48]         }
[49]     }
[50]     return (found ? r : 1e9);
[51] }
[52]
[53] signed main() {
[54]     ios_base::sync_with_stdio(0);
[55]     cin.tie(0);
[56]
```

```
[57]      string orig;
[58]      string want;
[59]      cin >> orig >> want;
[60]      n = orig.size();
[61]
[62]      vec<int> orig_freqmap(127);
[63]      orpref_freqs.resize(orig.size());
[64]
[65]      for (int i = 0; i < want.size(); ++i) {
[66]          ++want_freqmap[want[i]];
[67]      }
[68]
[69]      for (int i = 0; i < orig.size(); ++i) {
[70]          ++orig_freqmap[orig[i]];
[71]          orpref_freqs[i] = orig_freqmap;
[72]      }
[73]
[74]      int ansl = 0;
[75]      int anslen = 100'000;
[76]      for (int i = 0; i < orig.size(); ++i) {
[77]          int ans_at = search(i);
[78]          if (ans_at < anslen) {
[79]              anslen = ans_at;
[80]              ansl = i;
[81]          }
[82]      }
[83]
[84]      if (anslen == 100'000) {
[85]          exit(0);
[86]      }
[87]      // cout << ansl << " " << anslen << endl;
[88]      string res = orig.substr(ansl, anslen);
[89]
[90]      // string brute = "";
[91]      // for (int i = 0; i < orig.size(); ++i) {
[92]      //     for (int j = i; j < orig.size(); ++j) {
[93]      //         int len = j - i + 1;
[94]      //         string f = orig.substr(i, len);
[95]      //         vec<int> fmap(127);
[96]      //         for (int k = 0; k < f.size(); ++k) {
[97]      //             fmap[f[k]]++;
[98]      //         }
[99]      //         if ((brute == "" || brute.size() > len) &&
[100]     //             compat(want_freqmap, fmap)) {
[101]     //             brute = f;
[102]     //         }
[103]     //     }
[104]     // }
[105]     // if (res != brute) {
[106]     //     cout << "BAD!!!\n";
[107]     //     cout << res << endl;
[108]     //     cout << brute << endl;
[109]     // }
[110]
[111]     cout << res << endl;
[112] }
```

Посылка по задаче 6

```
[1] import sys
[2]
[3] tasks = {}
[4]
[5] for line in sys.stdin:
[6]     things = line.split(";")
[7]     name = things[0]
[8]     tests = things[1:]
[9]     tasks[name] = max(tasks.get(name, 0), len(tests))
[10]
[11] for key in sorted(tasks):
[12]     tst = tasks[key]
[13]     print(tst, tst - 1)
```