Олимпиада «Ломоносов» по информатике
2023-2024 учебный год. Заключительный тур
Работа участника с id заявки 1212675, логином inf24f_239

Сводный итог по всем задачам в проверяющей системе

Run ID	Time	User name	Problem	Language	Result	Tests	Score
467	3:09:07	inf24f_239	1	clang++	OK	28	100
470	3:09:41	inf24f_239	2	clang++	Partial solution	16	52
489	3:15:39	inf24f_239	3	clang++	Partial solution	27	96
503	3:19:18	inf24f_239	4	clang++	OK	21	100
513	3:22:23	inf24f_239	5	clang++	OK	22	100
550	3:32:29	inf24f_239	6	clang++	OK	11	100

548 технических баллов
91 итоговый балл

Посылка по задаче 1

```
[1] #pragma GCC target("avx2")
[2] #pragma GCC optimize("O3")
[3]
[4] #include <iostream>
[5] #include <algorithm>
[6] #include <cmath>
[7] #include <iomanip>
[8] #include <numeric>
[9] #include <vector>
[10]#include <set>
[11]#include <map>
[12]#include <unordered_set>
[13]#include <unordered_map>
[14]using namespace std;
[15]#define int long long
[16]#define Mod (1ll * 1000 * 1000 * 1000 + 7)
[17]//#define ll long long
[18]#define maxn 100000
[19]#define maxm 100000
[20]#define maxq 100000
[21]int n, q;
[22]int a[maxn + 1];
[23]int T[maxq + 1];
[24]struct segtree {
[25]    int lb, rb;
[26]    segtree *l = nullptr, *r = nullptr;
[27]    segtree() {}
[28]    segtree(int _lb, int _rb): lb(_lb), rb(_rb) {
[29]        if (lb < rb) {
[30]            l = new segtree(lb, (lb + rb) / 2); r = new segtree((lb + rb) / 2 + 1, rb);
[31]        }
[32]    }
[33]};
[34]segtree *root;
[35]int quant(int num, int i) {
[36]    if (i < 0 || num == 0) {
[37]        return 0;
[38]    }
[39]    if (num >= T[i]) {
[40]        return 1 + quant(num - T[i], i - 1);
[41]    }
[42]    return quant(num, i - 1);
[43]}
[44]signed main() {
[45]    ios_base::sync_with_stdio(false);
[46]    cin.tie(nullptr);
[47]    cin >> n;
[48]    T[0] = T[1] = T[2] = 0; T[3] = T[4] = 1;
[49]    int q = 4;
[50]    while (T[q] <= (1ll << 32)) {
[51]        T[q + 1] = T[q] + T[q - 1] + T[q - 2] + T[q - 3];
[52]        q++;
[53]    }
[54]    int ans = 0;
[55]    for (int i = 0; i < n; i++) {
[56]        cin >> a[i];
[57]        if (!(quant(a[i], q) % 2)) {
[58]            ans++;
[59]        }
[60]    }
[61]    cout << ans << endl;
[62]}
```

Посылка по задаче 2

```
[1]  #pragma GCC target("avx2")
[2]  #pragma GCC optimize("O3")
[3]
[4]  #include <iostream>
[5]  #include <algorithm>
[6]  #include <cmath>
[7]  #include <iomanip>
[8]  #include <numeric>
[9]  #include <string>
[10] #include <vector>
[11] #include <set>
[12] #include <map>
[13] #include <unordered_set>
[14] #include <unordered_map>
[15] using namespace std;
[16] #define int long long
[17] #define Mod (1ll * 1000 * 1000 * 1000 + 7)
[18] //#define ll long long
[19] #define maxn 500
[20] #define maxm 100000
[21] #define maxq 100000
[22] #define maxr 20
[23] #define maxc 200
[24] struct segtree {
[25]     int lb, rb;
[26]     segtree *l = nullptr, *r = nullptr;
[27]     segtree() {}
[28]     segtree(int _lb, int _rb): lb(_lb), rb(_rb) {
[29]         if (lb < rb) {
[30]             l = new segtree(lb, (lb + rb) / 2); r = new segtree((lb + rb) / 2 + 1, rb);
[31]         }
[32]     }
[33] };
[34] string str;
[35] map<char, int> id;
[36] map<int, char> ch;
[37] int arr[8][5000];
[38] int ind = 0;
[39] void add(int j, int l) {
[40]     int i = l;
[41]     while (arr[j][i] == 7) {
[42]         arr[j][i] = 0;
[43]         i--;
[44]     }
[45]     arr[j][i]++;
[46] }
[47] void evolvo(int l) {
[48]     if (id[str[ind]] == 8) {
[49]         ind++;
[50]         for (int i = 0; i < 8; i++) {
[51]             evolvo(l + 1);
[52]         }
[53]         return;
[54]     }
[55]     int j = id[str[ind]];
[56]     add(j, l);
[57]     ind++;
[58] }
[59] bool equal(int i, int j) {
[60]     int c = 0;
[61]     while (c < 5000 && arr[i][c] == arr[j][c]) {
[62]         c++;
[63]     }
[64]     return (c == 5000);
[65] }
[66] bool compare(int i, int j) {
[67]     int c = 0;
[68]     while (c < 5000 && arr[i][c] == arr[j][c]) {
[69]         c++;
[70]     }
[71]     if (c == 5000) {
[72]         return false;
[73]     }
```

```
[74]      return arr[i][c] < arr[j][c];
[75]  }
[76]  void print(int j) {
[77]      cout << ch[j] << endl;
[78]      if (arr[j][0] == 1) {
[79]          cout << "1.0" << endl;
[80]          return;
[81]      }
[82]      cout << "0.";
[83]      vector<int> vc;
[84]      for (int i = 1; i < 5000; i++) {
[85]          int c = arr[j][i];
[86]          vc.push_back((c & 4) > 0);
[87]          vc.push_back((c & 2) > 0);
[88]          vc.push_back((c & 1) > 0);
[89]      }
[90]      while (vc[vc.size() - 1] == 0) {
[91]          vc.pop_back();
[92]      }
[93]      for (int c : vc) {
[94]          if (c) {
[95]              cout << '1';
[96]          } else {
[97]              cout << '0';
[98]          }
[99]      }
[100]     cout << endl;
[101] }
[102] signed main() {
[103]     ios_base::sync_with_stdio(false);
[104]     cin.tie(nullptr);
[105]     id['Q'] = 8;
[106]     id['W'] = 7;
[107]     id['R'] = 6;
[108]     id['O'] = 5;
[109]     id['Y'] = 4;
[110]     id['G'] = 3;
[111]     id['C'] = 2;
[112]     id['B'] = 1;
[113]     id['V'] = 0;
[114]     ch[0] = 'V';
[115]     ch[1] = 'B';
[116]     ch[2] = 'C';
[117]     ch[3] = 'G';
[118]     ch[4] = 'Y';
[119]     ch[5] = 'O';
[120]     ch[6] = 'R';
[121]     ch[7] = 'W';
[122]     for (int i = 0; i < 5000; i++) {
[123]         for (int j = 0; j < 8; j++) {
[124]             arr[j][i] = 0;
[125]         }
[126]     }
[127]     cin >> str;
[128]     evolvo(0);
[129]     int i = 0;
[130]     for (int j = 1; j < 8; j++) {
[131]         if (compare(i, j)) {
[132]             i = j;
[133]         } else if (equal(i, j) && ch[i] - 'A' > ch[j] - 'A') {
[134]             i = j;
[135]         }
[136]     }
[137]     print(i);
[138] }
[139] /*
[140] Q, W, R, O, Y, G, C, B, V или D,
[141] */
[142] // QQDWWWWWWW      QWDWWWWWW     D D W W W W
[143] // QWROYGCBV
```

Посылка по задаче 3

```
[1]  #pragma GCC target("avx2")
[2]  #pragma GCC optimize("O3")
[3]
[4]  #include <iostream>
[5]  #include <algorithm>
[6]  #include <cmath>
[7]  #include <iomanip>
[8]  #include <numeric>
[9]  #include <string>
[10] #include <vector>
[11] #include <set>
[12] #include <map>
[13] #include <unordered_set>
[14] #include <unordered_map>
[15] using namespace std;
[16] #define int long long
[17] #define Mod (1ll * 1000 * 1000 * 1000 + 7)
[18] //#define ll long long
[19] #define maxn 500
[20] #define maxm 100000
[21] #define maxq 100000
[22] #define maxr 20
[23] #define maxc 200
[24] struct segtree {
[25]     int lb, rb;
[26]     segtree *l = nullptr, *r = nullptr;
[27]     segtree() {}
[28]     segtree(int _lb, int _rb): lb(_lb), rb(_rb) {
[29]         if (lb < rb) {
[30]             l = new segtree(lb, (lb + rb) / 2); r = new segtree((lb + rb) / 2 + 1, rb);
[31]         }
[32]     }
[33] };
[34] int n;
[35] string str[maxn];
[36] int dig[maxn][100];
[37] string get_digit(int num) {
[38]     if (num == 1) {
[39]         return "i";
[40]     } else if (num == 2) {
[41]         return "i(";
[42]     } else if (num == 3) {
[43]         return "i((";
[44]     } else if (num == 4) {
[45]         return "I";
[46]     } else if (num == 5) {
[47]         return "I(";
[48]     } else if (num == 6) {
[49]         return "I((";
[50]     } else if (num == 7) {
[51]         return "J";
[52]     } else if (num == 8) {
[53]         return "J)";
[54]     } else if (num == 9) {
[55]         return "J))";
[56]     } else if (num == 10) {
[57]         return "j";
[58]     } else if (num == 11) {
[59]         return "j)";
[60]     }
[61]     return "j))";
[62] }
[63] void convert(int i) {
[64]     for (int j = 0; j < 100; j++) {
[65]         dig[i][j] = 0;
[66]     }
[67]     if (str[i] == "[]") {
[68]         return;
[69]     }
[70]     int p = 0;
[71]     int len = (int)str[i].size();
[72]     int c = 0;
[73]     while (p < len) {
[74]         int q = p + 1;
[75]         while (q < len && str[i][q] == '(' || str[i][q] == ')') {
[76]             q++;
[77]         }
[78]         if (str[i][p] == 'i') {
[79]             dig[i][c] = 1;
[80]         } else if (str[i][p] == 'I') {
[81]             dig[i][c] = 4;
[82]         } else if (str[i][p] == 'J') {
[83]             dig[i][c] = 7;
```

```
[84]          } else if (str[i][p] == 'j') {
[85]              dig[i][c] = 10;
[86]          }
[87]          dig[i][c] += q - p - 1;
[88]          c++;
[89]          p = q;
[90]      }
[91]  }
[92]  bool compare(int i, int j) {
[93]      if (i == j) {
[94]          return false;
[95]      }
[96]      int c = 99;
[97]      while (c >= 0 && dig[i][c] == dig[j][c]) {
[98]          c--;
[99]      }
[100]     if (c < 0) {
[101]         return false;
[102]     }
[103]     return dig[i][c] < dig[j][c];
[104] }
[105] void printnum(int num) {
[106]     if (num > 144 + 12) {
[107]         int id = num / 144;
[108]         string sd = get_digit(id);
[109]         printnum(num % 144);
[110]         cout << sd;
[111]     } else if (num > 12) {
[112]         int id = num / 12;
[113]         string sd = get_digit(id);
[114]         printnum(num % 12);
[115]         cout << sd;
[116]     } else if (num > 0) {
[117]         int id = num;
[118]         string sd = get_digit(id);
[119]         cout << sd;
[120]     }
[121] }
[122] signed main() {
[123]     ios_base::sync_with_stdio(false);
[124]     cin.tie(nullptr);
[125]     cin >> n;
[126]     for (int i = 0; i < n; i++) {
[127]         cin >> str[i];
[128]         convert(i);
[129]     }
[130]     int mn = 0, mx = 0;
[131]     for (int i = 1; i < n; i++) {
[132]         if (!compare(mn, i)) {
[133]             mn = i;
[134]         }
[135]         if (!(compare(i, mx))) {
[136]             mx = i;
[137]         }
[138]     }
[139]     if (mn == mx) {
[140]         mx = n; mn = n - 1;
[141]     } else {
[142]         mn++; mx++;
[143]         if (mn > mx) {
[144]             swap(mn, mx);
[145]         }
[146]     }
[147]     printnum(mn);
[148]     cout << endl;
[149]     printnum(mx);
[150]     cout << endl;
[151] }
[152] /*
[153] 3
[154] J)j))ii
[155] J)j))ii
[156] J)j))ii
[157] */
[158] /*
[159] 4
[160] J)j))ii
[161] j))
[162] j))
[163] []
[164]
[165] */
```

Посылка по задаче 4

```
[1]  #pragma GCC target("avx2")
[2]  #pragma GCC optimize("O3")
[3]
[4]  #include <iostream>
[5]  #include <algorithm>
[6]  #include <cmath>
[7]  #include <iomanip>
[8]  #include <numeric>
[9]  #include <string>
[10] #include <vector>
[11] #include <set>
[12] #include <map>
[13] #include <unordered_set>
[14] #include <unordered_map>
[15] using namespace std;
[16] #define int long long
[17] #define Mod (1ll * 1000 * 1000 * 1000 + 7)
[18] //#define ll long long
[19] #define maxn 50000
[20] #define maxm 100000
[21] #define maxq 100000
[22] #define maxr 20
[23] #define maxc 200
[24] struct segtree {
[25]     int lb, rb;
[26]     segtree *l = nullptr, *r = nullptr;
[27]     segtree() {}
[28]     segtree(int _lb, int _rb): lb(_lb), rb(_rb) {
[29]         if (lb < rb) {
[30]             l = new segtree(lb, (lb + rb) / 2); r = new segtree((lb + rb) / 2 + 1, rb);
[31]         }
[32]     }
[33] };
[34]
[35] int r, c;
[36] int f[maxr][maxc];
[37] int quant[maxr][maxr][maxr][maxc];
[38] int getsum(int i1, int i2, int i3, int j) {
[39]     if (i1 == i2 and i2 == i3) {
[40]         return f[i1][j];
[41]     } else if (i1 == i2) {
[42]         return f[i1][j] + f[i3][j];
[43]     } else if (i2 == i3) {
[44]         return f[i1][j] + f[i2][j];
[45]     } else if (i3 == i1) {
[46]         return f[i2][j] + f[i3][j];
[47]     }
[48]     return f[i1][j] + f[i2][j] + f[i3][j];
[49] }
[50] int getval(int i1, int i2, int i3, int j) {
[51]     if (i1 < 0 || i1 >= r) {
[52]         return -1;
[53]     }
[54]     if (i2 < 0 || i2 >= r) {
[55]         return -1;
[56]     }
[57]     if (i3 < 0 || i3 >= r) {
[58]         return -1;
[59]     }
[60]     return quant[i1][i2][i3][j];
[61] }
[62] int getprev(int i1, int i2, int i3, int j) {
[63]     int mx = -1;
```

```
[64]     for (int d1 = -1; d1 <= 1; d1++) {
[65]         for (int d2 = -1; d2 <= 1; d2++) {
[66]             for (int d3 = -1; d3 <= 1; d3++) {
[67]                 mx = max(mx, getval(i1 + d1, i2 + d2, i3 + d3, j - 1));
[68]             }
[69]         }
[70]     }
[71]     return mx;
[72] }
[73] signed main() {
[74]     ios_base::sync_with_stdio(false);
[75]     cin.tie(nullptr);
[76]     cin >> r >> c;
[77]     int r1, r2, r3;
[78]     cin >> r1 >> r2 >> r3;
[79]     for (int i = 0; i < r; i++) {
[80]         for (int j = 0; j < c; j++) {
[81]             cin >> f[i][j];
[82]         }
[83]     }
[84]     for (int i1 = 0; i1 < r; i1++) {
[85]         for (int i2 = 0; i2 < r; i2++) {
[86]             for (int i3 = 0; i3 < r; i3++) {
[87]                 quant[i1][i2][i3][0] = -1;
[88]             }
[89]         }
[90]     }
[91]     quant[r1][r2][r3][0] = getsum(r1, r2, r3, 0);
[92]     int ans = -1;
[93]     for (int j = 1; j < c; j++) {
[94]         for (int i1 = 0; i1 < r; i1++) {
[95]             for (int i2 = 0; i2 < r; i2++) {
[96]                 for (int i3 = 0; i3 < r; i3++) {
[97]                     int mx = getprev(i1, i2, i3, j);
[98]                     quant[i1][i2][i3][j] = mx;
[99]                     if (mx != -1) {
[100]                        quant[i1][i2][i3][j] += getsum(i1, i2, i3, j);
[101]                        ans = max(ans, quant[i1][i2][i3][j]);
[102]                    }
[103]                }
[104]            }
[105]        }
[106]    }
[107]    cout << ans << endl;
[108] }
[109] /*
[110] 5 10
[111] 0 2 4
[112] 0 0 0 0  0 0 0 0 10 0
[113] 1 1 1 1  1 1 1 1 1  1
[114] 2 2 2 2  2 2 2 2 2  2
[115] 3 3 3 3  3 3 3 3 3  3
[116] 0 0 0 10 0 0 0 0 0  0
[117]
[118] */
```

Посылка по задаче 5

```
[1] #pragma GCC target("avx2")
[2] #pragma GCC optimize("O3")
[3]
[4] #include <iostream>
[5] #include <algorithm>
[6] #include <cmath>
[7] #include <iomanip>
[8] #include <numeric>
[9] #include <string>
[10]#include <vector>
[11]#include <set>
[12]#include <map>
[13]#include <unordered_set>
[14]#include <unordered_map>
[15]using namespace std;
[16]#define int long long
[17]#define Mod (1ll * 1000 * 1000 * 1000 + 7)
[18]//#define ll long long
[19]#define maxn 50000
[20]#define maxm 100000
[21]#define maxq 100000
[22]int n, m;
[23]int a[maxn + 1];
[24]std::string s;
[25]int pref[maxn + 1][100];
[26]std::string t;
[27]int quant[100];
[28]struct segtree {
[29]    int lb, rb;
[30]    segtree *l = nullptr, *r = nullptr;
[31]    segtree() {}
[32]    segtree(int _lb, int _rb): lb(_lb), rb(_rb) {
[33]        if (lb < rb) {
[34]            l = new segtree(lb, (lb + rb) / 2); r = new segtree((lb + rb) / 2 + 1, rb);
[35]        }
[36]    }
[37]};
[38]segtree *root;
[39]signed main() {
[40]    ios_base::sync_with_stdio(false);
[41]    cin.tie(nullptr);
[42]    cin >> s >> t;
[43]    n = (int)s.size(); m = (int)t.size();
[44]    for (int j = 0; j < 100; j++) {
[45]        pref[0][j] = 0;
[46]        quant[j] = 0;
[47]    }
[48]    for (int i = 1; i <= n; i++) {
[49]        for (int j = 0; j < 100; j++) {
[50]            pref[i][j] = pref[i - 1][j];
[51]        }
[52]        pref[i][s[i - 1] - 33]++;
[53]    }
[54]    for (int i = 0; i < m; i++) {
[55]        quant[t[i] - 33]++;
[56]    }
[57]    int len = n + 1, ind = -1;
[58]    int pointer[100];
[59]    for (int j = 0; j < 100; j++) {
[60]        pointer[j] = 0;
[61]    }
[62]    for (int i = 0; i < n; i++) {
[63]        int mx = i;
[64]        bool b = true;
[65]        for (int j = 0; j < 100 && b; j++) {
[66]            while (pointer[j] < i || (pointer[j] < n - 1 && pref[pointer[j] + 1][j] - pref[i]
[67][j] < quant[j])) {
[68]                pointer[j]++;
[69]            }
[70]            mx = max(mx, pointer[j]);
[71]            if (pref[pointer[j] + 1][j] - pref[i][j] < quant[j]) {
[72]                b = false;
[73]            }
[74]        }
[75]        if (b && mx - i + 1 < len) {
[76]            len = mx - i + 1;
[77]            ind = i;
[78]        }
[79]    }
[80]    if (len <= n) {
[81]        for (int j = 0; j < len; j++) {
[82]            cout << s[ind + j];
[83]        }
[84]        cout << endl;
[85]    } else {
[86]        cout << "" << endl;
[87]    }
    }
```

Посылка по задаче 6

```
[1]  #pragma GCC target("avx2")
[2]  #pragma GCC optimize("O3")
[3]
[4]  #include <iostream>
[5]  #include <algorithm>
[6]  #include <cmath>
[7]  #include <iomanip>
[8]  #include <numeric>
[9]  #include <string>
[10] #include <vector>
[11] #include <set>
[12] #include <map>
[13] #include <unordered_set>
[14] #include <unordered_map>
[15] using namespace std;
[16] #define int long long
[17] #define Mod (1ll * 1000 * 1000 * 1000 + 7)
[18] #define p 257
[19] //#define ll long long
[20] #define maxn 1000
[21] #define maxm 100000
[22] #define maxq 100000
[23] #define maxr 20
[24] #define maxc 200
[25] struct segtree {
[26]     int lb, rb;
[27]     segtree *l = nullptr, *r = nullptr;
[28]     segtree() {}
[29]     segtree(int _lb, int _rb): lb(_lb), rb(_rb) {
[30]         if (lb < rb) {
[31]             l = new segtree(lb, (lb + rb) / 2); r = new segtree((lb + rb) / 2 + 1, rb);
[32]         }
[33]     }
[34] };
[35] int n;
[36] map<string, int> id_task;
[37] string cs;
[38] int q[maxn];
[39] vector<vector<bool> > sub[maxn];
[40] void analysis(int ind) {
[41]     set<int> hash;
[42]     int quant = 0;
[43]     for (int i = 0; i < q[ind]; i++) {
[44]         quant = max(quant, (int)sub[ind][i].size());
[45]     }
[46]     for (int j = 0; j < quant; j++) {
[47]         int h = 0;
[48]         for (int i = 0; i < q[ind]; i++) {
[49]             h *= p;
[50]             if (h >= Mod) {
[51]                 h %= Mod;
[52]             }
[53]             if (sub[ind][i].size() > j) {
[54]                 h += sub[ind][i][j];
[55]             }
[56]             if (h >= Mod) {
[57]                 h -= Mod;
[58]             }
[59]         }
[60]         hash.insert(h);
[61]     }
[62]     cout << quant << ' ' << hash.size() << endl;
[63] }
[64] bool comp(pair<string, int> A, pair<string, int> B) {
[65]     int ind = 0;
[66]     while (ind < A.first.size() && ind < B.first.size() && A.first[ind] - 33 == B.first[ind] - 33) {
[67]         ind++;
[68]     }
[69]     if (ind >= A.first.size() && ind >= B.first.size()) {
[70]         return false;
[71]     }
```

```
[72]     if (ind >= A.first.size()) {
[73]         return true;
[74]     }
[75]     if (ind >= B.first.size()) {
[76]         return false;
[77]     }
[78]     return A.first[ind] - 33 < B.first[ind] - 33;
[79] }
[80] signed main() {
[81]     ios_base::sync_with_stdio(false);
[82]     cin.tie(nullptr);
[83]     n = 0;
[84]     while (getline(cin, cs)) {
[85]         if (cs.empty()) {
[86]             break;
[87]         }
[88]         string id;
[89]         int i = 0;
[90]         while (cs[i] != ';') {
[91]             id += cs[i];
[92]             i++;
[93]         }
[94]         i++;
[95]         int ind = -1;
[96]         if (id_task.count(id)) {
[97]             ind = id_task[id];
[98]         } else {
[99]             ind = n;
[100]            q[ind] = 0;
[101]            id_task[id] = n;
[102]            n++;
[103]        }
[104]        q[ind]++;
[105]        sub[ind].resize(q[ind]);
[106]        sub[ind][q[ind] - 1].clear();
[107]        while (i < cs.size()) {
[108]            if (i + 1 < cs.size() && cs[i] == 'O' && cs[i + 1] == 'K') {
[109]                sub[ind][q[ind] - 1].push_back(true);
[110]            } else {
[111]                sub[ind][q[ind] - 1].push_back(false);
[112]            }
[113]            while (i < cs.size() && cs[i] != ';') {
[114]                i++;
[115]            }
[116]            i++;
[117]        }
[118]    }
[119]    vector<pair<string, int> > an;
[120]    for (pair<string, int> pr : id_task) {
[121]        an.push_back(pr);
[122]    }
[123]    sort(an.begin(), an.end());
[124]    for (int ind = 0; ind < n; ind++) {
[125]        analysis(an[ind].second);
[126]    }
[127] }
[128] /*
[129] A;OK;OK;OK
[130] B;OK;OK;OK;OK
[131] A;OK;RT;RT
[132] B;OK;RT;OK
[133] A;OK;WA;RT
[134] B;OK;WA;WA;PE
[135]
[136]
[137] */
```