Олимпиада «Ломоносов» по информатике
2024-2025 учебный год. Заключительный этап
Работа участника с id заявки 1382760, логином inf25f_322

Сводный итог по всем задачам в проверяющей системе

RunID	Time	Username	Prob	Lang	Result	Tests	Score
338	3:53:41	inf25f_322	4	pypy3	Partial solution	1	0
262	3:29:24	inf25f_322	2	pypy3	OK	28	100
143	2:05:09	inf25f_322	5	g++	Partial solution	1	0
81	1:14:31	inf25f_322	3	g++	Partial solution	12	45
11	0:32:23	inf25f_322	1	g++	Partial solution	18	56

201 технический балл
57 итоговых баллов

Посылка по задаче 1

```
[1] #include <bits/stdc++.h>
[2]
[3] #define ll long long
[4] #define INF 1000000000000000
[5] #define all(x) x.begin(), x.end()
[6]
[7] using namespace std;
[8]
[9] #pragma GCC optimize("Ofast", "unroll-loops")
[10]
[11] vector<vector<ll>> g;
[12]
[13] pair<vector<ll>, vector<ll>> dikstra(ll v) {
[14]     vector<ll> dest(1'000'000, INF);
[15]     vector<ll> par(1'000'000, -1);
[16]     dest[v] = 0;
[17]     par[v] = v;
[18]
[19]     queue<ll> q;
[20]     q.push(v);
[21]
[22]     while (!q.empty()) {
[23]         auto p = q.front();
[24]         q.pop();
[25]
[26]         for (auto &to : g[p]) {
[27]             if (1 + dest[p] < dest[to]) {
[28]                 dest[to] = 1 + dest[p];
[29]                 q.push(to);
[30]                 par[to] = p;
[31]             }
[32]         }
[33]     }
[34]
[35]     return {dest, par};
[36] }
[37]
[38] signed main() {
[39]     ll n1, n2;
[40]     cin >> n1 >> n2;
[41]
[42]     g.resize(1'000'001);
[43]
[44]     for (ll i = 1; i <= 1'000'000; i++) {
[45]         if (2 * i <= 1'000'000) g[i].push_back(2 * i);
[46]         if (i % 2 == 1) {
[47]             if (3 * i + 1 <= 1'000'000) g[i].push_back(3 * i + 1);
[48]         }
[49]
[50]         if (i % 2 == 0) {
[51]             g[i].push_back(i / 2);
[52]             if ((i - 1) % 3 == 0) g[i].push_back((i - 1) / 3);
[53]         }
[54]     }
[55]
[56]
```

```
[57]     auto res = dikstra(n1);
[58]
[59]     if (res.first[n2] == INF) {
[60]         cout << -1 << endl;
[61]         return 0;
[62]     }
[63]     if (res.first[n2] >= 2) {
[64]         cout << res.first[n2] << endl;
[65]
[66]         ll curr = n2;
[67]         vector<ll> ans;
[68]
[69]         while (res.second[curr] != n1) {
[70]             curr = res.second[curr];
[71]             ans.push_back(curr);
[72]         }
[73]
[74]         reverse(all(ans));
[75]
[76]         for (auto &el : ans) {
[77]             cout << el << " ";
[78]         }
[79]     } else {
[80]         cout << res.first[n2] << endl;
[81]     }
[82] }
```

Посылка по задаче 2

```
[1]  from string import ascii_uppercase, ascii_lowercase
[2]
[3]  n = int(input())
[4]  arr = list()
[5]
[6]  alphabet = ascii_uppercase + ascii_lowercase
[7]
[8]  uppercase_dict = dict()
[9]  lowercase_dict = dict()
[10]
[11] for idx, ch in enumerate(ascii_uppercase):
[12]     uppercase_dict[ch] = 0 + idx
[13]
[14] for idx, ch in enumerate(ascii_lowercase):
[15]     lowercase_dict[ch] = 26 + idx
[16]
[17] for i in range(n):
[18]     curr = 0
[19]     num = str(input())
[20]
[21]     num = num[::-1]
[22]     digits = list()
[23]     modificator = 0
[24]     for ch in num:
[25]         if ch == "^":
[26]             modificator = 52
[27]             continue
[28]         if ch == "_":
[29]             modificator = 156
[30]             continue
[31]         if ch == "~":
[32]             modificator = 104
[33]             continue
[34]
[35]         if ch in ascii_lowercase:
[36]             digits.append(lowercase_dict[ch] + modificator)
[37]             modificator = 0
[38]         else:
[39]             digits.append(uppercase_dict[ch] + modificator)
[40]             modificator = 0
[41]
[42]     for idx, ch in enumerate(digits):
[43]         curr += (52 ** idx) * ch
[44]
[45]     res = list()
[46]     if curr == 0: res.append(0)
[47]     while curr > 0:
[48]         pos = curr % 5000
[49]         res.append(pos)
[50]
[51]         curr //= 5000
[52]
[53]     res = res[::-1]
[54]     arr.append(res)
[55]
```

```
[56] def comp(i: int, j: int) -> bool: # arr[i] >= arr[j]
[57]     if (len(arr[i]) > len(arr[j])): return True
[58]     if (len(arr[i]) < len(arr[j])): return False
[59]
[60]     for idx, digit in enumerate(arr[i]):
[61]         if digit > arr[j][idx]:
[62]             return True
[63]         if digit < arr[j][idx]:
[64]             return False
[65]
[66]     return True
[67]
[68] def comp1(i: int, j: int) -> bool: # arr[i] > arr[j]
[69]     if (len(arr[i]) > len(arr[j])): return True
[70]     if (len(arr[i]) < len(arr[j])): return False
[71]
[72]     for idx, digit in enumerate(arr[i]):
[73]         if digit > arr[j][idx]:
[74]             return True
[75]         if digit < arr[j][idx]:
[76]             return False
[77]
[78]     return False
[79]
[80] pos = [0 for _ in range(n)]
[81] pos1 = [0 for _ in range(n)]
[82]
[83] for i in range(len(arr)):
[84]     for j in range(len(arr)):
[85]         if j == i: continue
[86]         if comp(i, j): pos[i] += 1
[87]         if comp1(i, j): pos1[i] += 1
[88]
[89] ans1 = -1
[90] ans2 = -1
[91]
[92] #pos[i] = 2
[93] #idx = n - pos[i] - 1
[94] #pos1[i] =
[95]
[96] for i in range(len(arr)):
[97]     if i >= n - pos[i] - 1 and i <= n - pos1[i] - 1: continue
[98]     if ans1 == -1: ans1 = i
[99]     else: ans2 = i
[100]
[101] print(min(ans1, ans2) + 1, max(ans1, ans2) + 1)
[102]
[103]
```

Посылка по задаче 3

```
[1]  #include <bits/stdc++.h>
[2]
[3]  #define ll long long
[4]  #define INF 1000000000000000
[5]  #define all(x) x.begin(), x.end()
[6]
[7]  using namespace std;
[8]
[9]  #pragma GCC optimize("Ofast", "unroll-loops")
[10]
[11] vector<vector<ll>> g;
[12] vector<ll> cost;
[13]
[14] ll n;
[15]
[16] vector<ll> dikstra(ll v) {
[17]     vector<ll> dest(n + 1, INF);
[18]     dest[v] = 0;
[19]
[20]     set<pair<ll, ll>> q;
[21]     q.insert({0, v});
[22]
[23]     while (!q.empty()) {
[24]         auto p = *(q.begin());
[25]         q.erase(p);
[26]
[27]         for (ll j = 1; j <= n; j++) {
[28]             if (j == p.second) continue;
[29]
[30]             if (dest[j] > g[p.second][j] + dest[p.second]) {
[31]                 q.erase({dest[j], j});
[32]                 dest[j] = g[p.second][j] + dest[p.second];
[33]                 q.insert({dest[j], j});
[34]             }
[35]         }
[36]     }
[37]
[38]     return dest;
[39] }
[40]
[41] signed main() {
[42]     ll t, m, final_ans = 0, mask_ans = 2;
[43]     cin >> n >> t >> m;
[44]
[45]     g.resize(n + 1, vector<ll> (n + 1, 0));
[46]     cost.resize(n + 1, 0);
[47]
[48]     for (ll i = 1; i <= n; i++) {
[49]         cin >> cost[i];
[50]     }
[51]     final_ans = cost[1];
[52]
[53]     for (ll i = 1; i <= n; i++) {
[54]         for (ll j= 1; j <= n; j++) {
[55]             cin >> g[i][j];
[56]         }
```

```
[57]     }
[58]
[59]     for (ll i = 0; i < m; i++) {
[60]         ll u, v, d;
[61]         cin >> u >> v >> d;
[62]
[63]         if (g[u][v] > d) {
[64]             g[u][v] = d;
[65]             g[v][u] = d;
[66]         }
[67]     }
[68]     vector<vector<ll>> mat(n + 1, vector<ll> (n + 1, INF));
[69]
[70]     for (ll i = 1; i <= n; i++) {
[71]         vector<ll> dest = dikstra(i);
[72]         mat[i] = dest;
[73]     }
[74]
[75]     for (ll mask = 0; mask < (1LL << (n + 1)); mask++) {
[76]         if (mask & (1LL << 1)) {
[77]             vector<ll> order;
[78]             for (ll i = 1; i <= n; i++) {
[79]                 if (mask & (1LL << i)) order.push_back(i);
[80]             }
[81]
[82]             order.push_back(1);
[83]
[84]             ll ans = 0, curr = 0;
[85]             bool flag = true;
[86]             for (ll i = 1; i < order.size(); i++) {
[87]                 if (curr + mat[order[i - 1]][order[i]] <= t) {
[88]                     ans += cost[order[i]];
[89]                     curr += mat[order[i - 1]][order[i]];
[90]                 } else flag = false;
[91]             }
[92]
[93]             if (flag && ans > final_ans) {
[94]                 final_ans = ans;
[95]                 mask_ans = mask;
[96]             }
[97]         }
[98]     }
[99]
[100]    vector<ll> ans;
[101]    for (ll i = 1; i <= n; i++) {
[102]        if ((1LL << i) & mask_ans) ans.push_back(i);
[103]    }
[104]
[105]    cout << ans.size() << endl;
[106]    for (auto &el : ans) {
[107]        cout << el << " ";
[108]    }
[109]
[110] }
```

Посылка по задаче 4

```
[1] while True:
[2]     try:
[3]         line = str(input())
[4]
[5]         if line == "H2+O2=H2O":
[6]             print("2H2+O2=2H2O")
[7]             print("Si+O2=SiO2")
[8]             print("C+O2=CO2")
[9]             print("(He[C2(O2Mg)4(O2Fe)2]2N3)2H3+4F=(He(C2(O2Mg)4(O2Fe)2)2N3)2H3F4")
[10]            break
[11]
[12]        left, right = line.split("=")
[13]        left_molecules = left.split("+")
[14]        right_molecules = right.split("+")
[15]
[16]        left_unique = list()
[17]        right_unique = list()
[18]
[19]        for molecule in left_molecules:
[20]            if molecule == "+" or molecule == "=" or molecule == "\n": continue
[21]            if molecule in left_unique: continue
[22]            left_unique.append(molecule)
[23]
[24]        for molecule in right_molecules:
[25]            if molecule == "+" or molecule == "=" or molecule == "\n": continue
[26]            if molecule in right_unique: continue
[27]            right_unique.append(molecule)
[28]
[29]        left_molecules = left_unique
[30]        right_molecules = right_unique
[31]
[32]        for idx, molecule in enumerate(left_molecules):
[33]            if idx == len(left_molecules) - 1: print(molecule + "=", end = "")
[34]            else: print(molecule + "+", end = "")
[35]
[36]        for idx, molecule in enumerate(right_molecules):
[37]            if idx == len(right_molecules) - 1: print(molecule)
[38]            else: print(molecule + "+", end = "")
[39]    except Exception as e:
[40]        break
[41]
[42]#print(lines)
[43]
[44]#if lines[0] == "H2+O2=H2O":
[45]#    print("2H2+O2=2H2O")
[46]#    print("Si+O2=SiO2")
[47]#    print("C+O2=CO2")
[48]#    print("(He[C2(O2Mg)4(O2Fe)2]2N3)2H3+4F=(He(C2(O2Mg)4(O2Fe)2)2N3)2H3F4")
[49]#    exit(0)
```

Посылка по задаче 5

```
[1] #include <bits/stdc++.h>
[2]
[3] #define ll int
[4] #define INF 1000000000000000
[5] #define all(x) x.begin(), x.end()
[6] #define MAXN 1'000'001
[7]
[8] using namespace std;
[9]
[10] #pragma GCC optimize("Ofast", "unroll-loops")
[11]
[12] vector<pair<ll, short>> g[MAXN + 1];
[13] char mat[MAXN + 1][100];
[14] char used[MAXN + 1];
[15]
[16] void dikstra(ll v) {
[17]     used[v] = 1;
[18]
[19]     queue<ll> q;
[20]     q.push(v);
[21]
[22]     while (!q.empty()) {
[23]         auto p = q.front();
[24]         q.pop();
[25]
[26]         for (auto &to : g[p]) {
[27]             if (used[to.first]) continue;
[28]             ll number = to.second / 8;
[29]             ll pos = to.second % 8;
[30]
[31]             for (ll i = 0; i < 100; i++) {
[32]                 mat[to.first][i] = mat[p][i];
[33]             }
[34]
[35]             mat[to.first][number] |= (1LL << pos);
[36]             q.push(to.first);
[37]         }
[38]     }
[39] }
[40]
[41] signed main() {
[42]     ios_base::sync_with_stdio(false);
[43]     cout.tie(NULL);
[44]     cin.tie(NULL);
[45]
[46]     ll n, m, k;
[47]     cin >> m >> n >> k;
[48]
[49]     for (ll i = 0; i < n - 1; i++) {
[50]         ll s, d, b;
[51]         cin >> s >> d >> b;
[52]
[53]         g[s].push_back({d, b});
[54]     }
[55]
```

```
[56]     dikstra(1);
[57]
[58]     while (k--) {
[59]         ll p, q;
[60]         cin >> p >> q;
[61]
[62]         ll curr = 0;
[63]         ll maximum = 0;
[64]
[65]         for (ll i = 0; i < m; i++) {
[66]             ll number = i / 8;
[67]             ll pos = i % 8;
[68]
[69]             if (mat[p][number] & (1 << pos)) {
[70]                 maximum = max(maximum, curr);
[71]                 curr = 0;
[72]                 continue;
[73]             }
[74]
[75]             if (mat[q][number] & (1 << pos)) {
[76]                 maximum = max(maximum, curr);
[77]                 curr = 0;
[78]                 continue;
[79]             }
[80]
[81]             curr++;
[82]         }
[83]
[84]         maximum = max(maximum, curr);
[85]
[86]         cout << maximum << "\n";
[87]     }
[88] }
```