Олимпиада «Ломоносов» по информатике
2024-2025 учебный год. Заключительный этап
Работа участника с id заявки 1400563, логином inf25f_313

Сводный итог по всем задачам в проверяющей системе

RunID	Time	Username	Prob	Lang	Result	Tests	Score
329	3:51:07	inf25f_313	3	g++	Partial solution	22	95
259	3:26:00	inf25f_313	4	python3	Partial solution	0	0
102	1:32:50	inf25f_313	1	g++	Partial solution	13	36
22	0:37:27	inf25f_313	2	python3	OK	28	100
N/A	N/A	inf25f_313	5	N/A	N/A	0	0

231 технический балл
66 итоговых баллов

Посылка по задаче 1

```
[1] #include <iostream>
[2] #include <vector>
[3] #include <cstdint>
[4] using namespace std;
[5]
[6] #define int long long
[7]
[8] const int N = 100000;
[9] int ans = (int) 1e9;
[10] vector<int> ans_path;
[11] int n2;
[12] int c = 0;
[13]
[14] void rec(int cur, int pr, int steps, vector<int> path) {
[15]     path.push_back(cur);
[16]     c++;
[17]     if (c == N) {
[18]         cout << -1;
[19]         exit(0);
[20]     }
[21]     if (cur > N || steps > ans) {
[22]         return;
[23]     }
[24]     if (cur == n2) {
[25]         if (steps < ans) {
[26]             ans = steps;
[27]             ans_path.clear();
[28]             for (int e : path) {
[29]                 ans_path.push_back(e);
[30]             }
[31]         }
[32]         return;
[33]     }
[34]     if (cur % 2 == 0) {
[35]         if (cur * 2 != pr) {
[36]             rec(cur * 2, cur, steps + 1, path);
[37]         }
[38]         if (cur / 2 != pr) {
[39]             rec(cur / 2, cur, steps + 1, path);
[40]         }
[41]         if ((cur - 1) % 3 == 0 && pr != (cur - 1) / 3) {
[42]             rec((cur - 1) / 3, cur, steps + 1, path);
[43]         }
[44]     } else {
[45]         if (cur * 2 != pr) {
[46]             rec(cur * 2, cur, steps + 1, path);
[47]         }
[48]         if (cur * 3 + 1 != pr) {
[49]             rec(cur * 3 + 1, cur, steps + 1, path);
[50]         }
[51]     }
[52] }
[53]
[54] int32_t main() {
[55]     int n1;
[56]     cin >> n1 >> n2;
[57]     rec(n1, -1, 0, {});
[58]     cout << ans << "\n";
[59]     if (ans > 1) {
[60]         for (int i = 1; i < ans_path.size() - 1; i++) {
[61]             cout << ans_path[i];
[62]             if (i != ans_path.size() - 2) {
[63]                 cout << " ";
[64]             }
[65]         }
[66]     }
[67] }
```

Посылка по задаче 2

```
[1] alph = "ABCDEFGHIJKLMNOPQRSTUVWXYZ" + "ABCDEFGHIJKLMNOPQRSTUVWXYZ".lower()
[2] d1 = {}
[3] d2 = {"^": 52, "~": 104, "_": 156}
[4] d2ks = "^~_"
[5]
[6]
[7] def filld2():
[8]     for i in range(52):
[9]         d1[alph[i]] = i
[10]
[11]
[12]def to10(n):
[13]    # n - list
[14]    n10 = 0
[15]    for i in range(len(n)):
[16]        n10 += n[i] * (52 ** (len(n) - i - 1))
[17]    return n10
[18]
[19]
[20]def main():
[21]    filld2()
[22]
[23]    n = int(input())
[24]    a = []
[25]    for _ in range(n):
[26]        number = []
[27]        s = input()
[28]        i = 0
[29]        while i < len(s):
[30]            if i + 1 < len(s):
[31]                if s[i + 1] in d2ks:
[32]                    number.append(d1[s[i]] + d2[s[i + 1]])
[33]                    i += 1
[34]                else:
[35]                    number.append(d1[s[i]])
[36]            else:
[37]                number.append(d1[s[i]])
[38]            i += 1
[39]        # print(number)
[40]        a.append(to10(number))
[41]
[42]    for i in range(1, n):
[43]        if a[i] > a[i - 1]:
[44]            ind = i - 1
[45]            break
[46]
[47]    for i in range(n):
[48]        if i != ind:
[49]            q = a[i]
[50]            w = a[ind]
[51]            a[i] = a[ind]
[52]            a[ind] = q
[53]            if a == sorted(a, reverse=True):
[54]                print(ind + 1, i + 1)
[55]                break
[56]            a[i] = q
[57]            a[ind] = w
[58]
[59]
[60]if __name__ == "__main__":
[61]    main()
```

Посылка по задаче 3

```
[1]  #include <iostream>
[2]  #include <vector>
[3]  #include <set>
[4]  #include <queue>
[5]  #include <cstdint>
[6]  using namespace std;
[7]
[8]  #define int long long
[9]
[10] struct node{
[11]     int v, d;
[12] };
[13]
[14] const int N = 11, INF = INT64_MAX;
[15] int w[N];
[16] vector<node> g[N];
[17] int T;
[18]
[19] set<int> answer;
[20] int mx = -10000000;
[21]
[22] int tme[N];
[23] void dkstr() {
[24]     queue<node> q;
[25]
[26]     for (int i = 0; i < N; i++) {
[27]         tme[i] = INF;
[28]     }
[29]     tme[0] = 0;
[30]
[31]     q.push({0, 0});
[32]     while (!q.empty()) {
[33]         node qi = q.front();
[34]         q.pop();
[35]         for (node nbr : g[qi.v]) {
[36]             if (tme[nbr.v] > qi.d + nbr.d) {
[37]                 tme[nbr.v] = qi.d + nbr.d;
[38]                 q.push({nbr.v, qi.d + nbr.d});
[39]             }
[40]         }
[41]     }
[42] }
[43]
[44] void dfs(int v, int t, set<int> ans, int res, int c) {
[45]     c += 1;
[46]     if (c > 1000) {  // ответ точно не изменится
[47]         return;
[48]     }
[49]     bool fl = true;
[50]     for (int e : ans) {
[51]         if (v == e) {
[52]             fl = false;
[53]             break;
[54]         }
[55]     }
[56]     if (fl) {
[57]         res += w[v];
[58]         ans.insert(v);
[59]     }
[60]     if (T - t >= tme[v]) {  // то есть из текущей вершины можно успеть обратно
[61]         if (res == mx) {
[62]             if (answer.size() > ans.size()) {
[63]                 answer = ans;
[64]             } else if (answer.size() == ans.size() && ans < answer) {
[65]                 answer = ans;
[66]             } else {
[67]                 return;
[68]             }
[69]         } else if (res > mx) {
[70]             mx = res;
[71]             answer = ans;
[72]         }
[73]     } else {
[74]         return;
[75]     }
```

```
[76]
[77]     for (node e : g[v]) {
[78]         dfs(e.v, t + e.d, ans, res, c);
[79]     }
[80] }
[81]
[82] int32_t main() {
[83]     // input
[84]     int n, m;
[85]     cin >> n >> T >> m;
[86]     for (int i = 0; i < n; i++) {
[87]         cin >> w[i];
[88]     }
[89]     for (int i = 0; i < n; i++) {
[90]         for (int j = 0; j < n; j++) {
[91]             int d;
[92]             cin >> d;
[93]             if (i != j) {
[94]                 g[i].push_back({j, d});
[95]             }
[96]         }
[97]     }
[98]     for (int i = 0; i < m; i++) {
[99]         int u, v, d;
[100]        cin >> u >> v >> d;
[101]        --u; --v;
[102]        for (auto & e : g[u]) {
[103]            if (e.v == v && e.d > d) {
[104]                e.d = d;
[105]            }
[106]        }
[107]        for (auto & e : g[v]) {
[108]            if (e.v == u && e.d > d) {
[109]                e.d = d;
[110]            }
[111]        }
[112]    }
[113]    //
[114]
[115]    dkstr();
[116]    dfs(0, 0, {}, 0, 0);
[117]
[118]    cout << answer.size() << "\n";
[119]    int k = 0;
[120]    for (int e : answer) {
[121]        k++;
[122]        if (k != answer.size()) {
[123]            cout << e + 1 << " ";
[124]        } else {
[125]            cout << e + 1;
[126]        }
[127]    }
[128] }
```

Посылка по задаче 4

```
[1] from sys import stdin
[2]
[3] a = list(i[:-1] for i in stdin)
[4]
[5] for i in a:
[6]     print(i)
```

Посылка по задаче 5
Посылок по задаче 5 не было отправлено.