

Разбор задач школьного этапа всероссийской
олимпиады школьников по программированию.
9-11 классы.

18 сентября 2018

Ответственные за подготовку тура

1 Организационные моменты

- 1 Егоров Алексей Леонидович

2 Подготовка задач

- 1 Кучеренко Демид
- 2 Ведерников Николай
- 3 Трубецкой Богдан
- 4 Коробков Роман
- 5 Шпаковский Денис
- 6 Подуреманных Илья

Задача А "Расшифровать рэперов"

- Сделаем то, что просят в задаче
- Будем идти по строке и, если, встретили классную букву выведем ее
- Повторим то же для согласных
- Для проверки на гласность буквы удобно использовать множества
- Решение работает за $O(len(s))$

Example (Авторское решение)

```
s = input()
q = {'a', 'e', 'i', 'o', 'u', 'y'}
for i in s:
    if (i in q):
        print(i, end = "")
for i in s:
    if (not i in q and i != ' '):
        print(i, end = "")
```

Задача В "Математические трюки"

- Для решения задачи смоделируем ее
- Увеличивая и числитель и знаменатель на один значение дроби тоже увеличивается, так как $a < b$
- Будем увеличивать и числитель и знаменатель на один и сокращать, используя улучшенный алгоритм Евклида, пока первая дробь меньше второй
- Решение работает за $\mathcal{O}(a * \log(a))$
- Решение можно улучшить применив бинарный поиск

Задача В "Математические трюки"

Example (Авторское решение)

```
def gcd(a, b):
    while b != 0:
        a, b = b, a % b
    return a
a = int(input())
b = int(input())
c = int(input())
d = int(input())
ans = 0
while a * d < c * b:
    a += 1
    b += 1
    m = gcd(a, b)
    a //= m
    b //= m
    ans += 1
if a == c and b == d:
    print(ans)
else:
    print(0)
```

Задача C "СУ-ДО-КУ"

- В задаче требуется написать то, что требуется
- Проверим, что в каждой строчке встречаются все числа от 1 до 9
- Также проверим все столбы и специальные квадраты

Задача С "СУ-ДО-КУ"

Example (Авторское решение)

```
const int MAXN = (int)10;
int t[MAXN][MAXN];
bool ok_line(size_t line) {
    vector<bool> was(9);
    for (size_t i = 0; i < 9; i++) was[t[line][i] - 1] = true;
    return std::all_of(was.begin(), was.end(), [](bool i){return i;});
}
bool ok_column(size_t column) {
    vector<bool> was(9);
    for (size_t i = 0; i < 9; i++) was[t[i][column] - 1] = true;
    return std::all_of(was.begin(), was.end(), [](bool i){return i;});
}
bool ok_square(size_t pos) {
    vector<bool> was(9);
    size_t start_i = (pos / 3) * 3;
    size_t start_j = (pos % 3) * 3;
    for (size_t i = 0; i < 3; i++) {
        for (size_t j = 0; j < 3; j++) {
            was[t[start_i + i][start_j + j] - 1] = true;
        }
    }
    return std::all_of(was.begin(), was.end(), [](bool i){return i;});
}
```

Example (Авторское решение)

```
int main() {
    for (int i = 0; i < 9; i++) {
        for (int j = 0; j < 9; j++) {
            cin >> t[i][j];
        }
    }
    bool ok = true;
    for (size_t i = 0; i < 9; i++) {
        ok &= ok_square(i) && ok_column(i) && ok_line(i);
    }
    if (ok) cout << "Yes\n";
    else cout << "No\n";
    return 0;
}
```


Задача D "А кто не любит?"

- Заметим, что смысл ложиться спать есть только в тот день, когда к старцу кто-то приходит
- Теперь переберем день, в который мы будем ложиться спать
- Пересчитаем ответ
- Решение за $\mathcal{O}(n \cdot \log(n))$

Задача D "А кто не любит?"

Example (Авторское решение)

```
int n, k, T;
int a[200007];
int main(){
    cin >> n >> k >> T;
    k++;
    for (int i = 0; i < n; i++) cin >> a[i];
    sort(a, a + n);
    if (n < k) cout << T, exit(0);
    int ans = min(T, a[k - 1]);
    for (int i = 0; i < n; i++) {
        int start = a[i];
        if (start >= T) break;
        if (i + k >= n) ans = max(ans, T - start);
        else {
            int finish = min(T, a[i + k]);
            ans = max(ans, finish - start);
        }
    }
    cout << ans;
}
```

Задача E "Опять в школу"

- В задаче требуется для клетки таблицы находить ближайший элемент по манхеттенскому расстоянию с некоторым условием.
- Расстояние = $|x_1 - x_2| + |y_1 - y_2|$
- Раскроем эти модули для 4-х случаев. В каждом из них нужно независимо выбрать максимум по $(x + y)$
- Решение за $\mathcal{O}(n \cdot m)$

Задача E "Опять в школу"

Example (Авторское решение)

```
const int dd = 1507;
short int A[dd][dd];
short int T1[dd], T2[dd], T3[dd], T4[dd];
void upd(short int &a, short int b) {
    if (a < b) a = b;
}
int main() {
    int n, m, k;
    cin >> n >> m >> k;
    for (int i = 0; i < n; i++) for (int j = 0; j < m; j++) {
        cin >> A[i][j];
        A[i][j]--;
    }
    fill(T1, T1 + k, -20000);
    fill(T2, T2 + k, -20000);
    fill(T3, T3 + k, -20000);
    fill(T4, T4 + k, -20000);
```

Задача E "Опять в школу"

Example (Авторское решение)

```
for (int i = 0; i < n; i++) for (int j = 0; j < m; j++) {
    short int c = A[i][j];
    upd(T1[c], i + j);
    upd(T2[c], i - j);
    upd(T3[c], -i + j);
    upd(T4[c], -i - j);
}
for (int c = k - 2; c >= 0; c--) {
    upd(T1[c], T1[c + 1]);
    upd(T2[c], T2[c + 1]);
    upd(T3[c], T3[c + 1]);
    upd(T4[c], T4[c + 1]);
}
for (int i = 0; i < n; i++) for (int j = 0; j < m; j++) {
    short int cur = 0;
    short int c = A[i][j];
    upd(cur, abs(T1[c] - (i + j)));
    upd(cur, abs(T2[c] - (i - j)));
    upd(cur, abs(T3[c] - (-i + j)));
    upd(cur, abs(T4[c] - (-i - j)));
    cout << cur << " \n"[j == m - 1];
}
return 0;
}
```

Все просто!