Заключительный этап

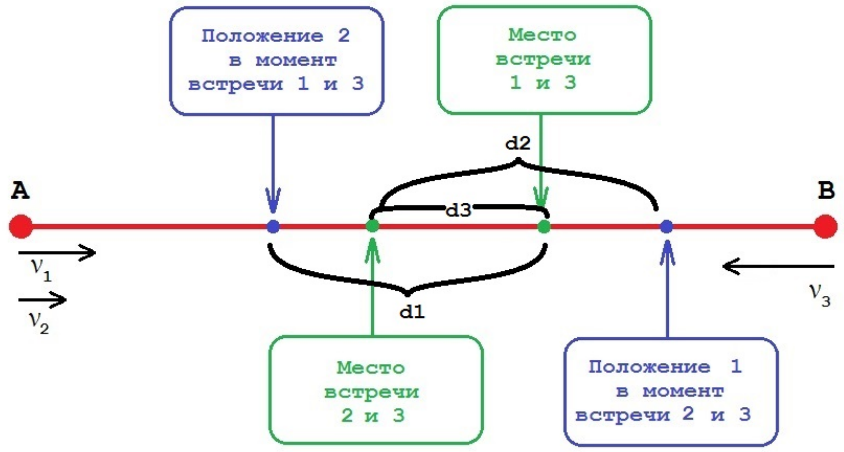
Индивидуальный предметный тур

Информатика. 8–11 класс

Задача III.1.1.1. Интеллектуальная ГИС (12 баллов)

Вы являетесь тестером в команде разработчиков новой интеллектуальной геоинформационной системы. Данная система помимо прямых расчетов расположения отслеживаемых объектов, должна уметь определять необходимые данные (расстояния, координаты и т. п.) путем сопоставления множества косвенных данных.

В данный момент тестируется следующая ситуация: есть три объекта 1, 2 и 3. Объекты 1 и 2 одновременно стартуют из точки A в направлении точки B, объект 3 в этот же момент стартует из точки B в направлении точки A. Известно, что скорость объекта 1 (обозначим ее v_1) строго больше скорости объекта 2 (обозначим ее v_2), но сами значения этих скоростей не известны. Однако известна их разность $dv = v_1 - v_2$. Очевидно, что при своем движении, объект 3 сначала встретится с объектом 1, а затем через какое-то время встретится с объектом 2. Известны следующие расстояния: d_1 — расстояние между 1 и 2 в момент встречи третьего и первого, d_2 — расстояние между 1 и 2 в момент встречи третьего и второго, d_3 — расстояние между точками встречи. Требуется по заданным четырем параметрам определить расстояние между точками A и B. В момент встречи второго и третьего первый не успевает достичь точки B.



Формат входных данных

На вход подаются четыре положительных числа dv, d_1, d_2, d_3 через пробел. Все числа целые в пределах от 1 до 10^9. $d_3 < d_1 < d_2$. Входные данные таковы, что в момент встречи второго и третьего первый еще не успевает достичь точки A.

Формат выходных данных

Вывести одно число — расстояние между точками AB. Ответ следует выводить с округлением ровно до пяти знаков после десятичной точки.

Примеры

Пример №1

Стандартный ввод
3 9 12 5

Стандартный вывод
36.00000

Пример программы-решения

Ниже представлено решение на языке C++.

```cpp
#include <bits/stdc++.h>
#define int long long
using namespace std;

int main(){
    long double dv, d1, d2, d3;
    cin >> dv >> d1 >> d2 >> d3;

    cout << fixed << setprecision(5) << d1 * d2 / (d2 - d1);
    return 0;
}
```

Пример программы-решения

Ниже представлено решение на языке Python.

```python
rom decimal import Decimal
a = list(map(int, input().split()))
d1 = Decimal(str(a[1]))
d2 = Decimal(a[2])
n = 5
a = d1 * d2 / (d2 - d1)
print(f'{a:.{n}f}')
```

Задача III.1.1.2. Самоорганизация в городе (15 баллов)

Рассмотрим следующую модель обмена ресурсами внутри некоторого города. Город состоит из n расположенных в один ряд кварталов. Каждый квартал изначально имеет a_i единиц запаса некоторого ресурса, причем никакие два квартала не обладают одинаковыми запасами. Для любых двух рядом расположенных кварталов возможна операция передачи одной единицы ресурса от большего по запасам к меньшему, но, само собой, отдающий квартал не должен стать беднее по запасам того, кому он отдает. Основным требованием властей города является разнообразие запасов, что означает что ни в какой момент времени никакие два квартала не должны обладать одинаковыми по величине запасами. Таким образом, операция передачи может быть осуществлена только тогда, когда после ее выполнения новые величины запасов обоих участвовавших в операции передачи ресурса кварталов не совпадают ни с одним другим кварталом (в том числе и между собой). Если есть несколько пар рядом стоящих кварталов, имеющих возможность передачи, то выбирается пара с наибольшей разностью, а если и таких несколько, то среди них выбирается пара с минимальным значением запаса у получающего помощь квартала. За одну единицу времени возможен ровно один обмен. Такие операции поочередно производятся до тех пор, пока это не нарушает правила разнообразия. Требуется выяснить, каким станет распределение ресурса после окончания обменов.

Формат входных данных

В первой строке содержится число n — количество кварталов. Во второй строке находятся n натуральных чисел a_i через пробел — запасы ресурса у соответствующего квартала. Все величины запасов попрано различны. Все числа на входе в пределах от 1 до 1000.

Формат выходных данных

Вывести в одну строку через пробел n чисел — запасы в кварталах после того, как станет невозможно производить операции передачи ресурса.

Примеры

Пример №1

Стандартный ввод
`6` `2 10 9 3 6 12`

Стандартный вывод
`3 9 8 5 7 10`

Пояснение к примеру 1

Приведем последовательность перераспределений:
0) 2 10 **9 3** 6 12
1) **2 10** 8 4 6 12
2) 3 9 8 4 **6 12**
3) 3 9 8 **4 7** 11
4) 3 9 8 5 **6 11**
5) 3 9 8 5 7 10

В нулевой строке исходные значения. Самая большая разница между 10 и 2, но эту операцию запрещено проводить, так как тогда во втором квартале будет 9, но 9 уже есть в третьем. Следующая по величине разница между 9 и 3, а так же между 12 и 6. Тогда производится помощь наиболее нуждающемуся в этих двух вариантах, а именно $9 \to 3$. В первой строке результат передачи. Теперь самая большая разница между 2 и 10 и ничто не мешает произвести передачу $10 \to 2$. Во второй строке результат. Теперь снова есть два варианта $9 \to 3$ и $12 \to 6$, но теперь первый запрещен по причине того, что после этого получатся числа 4 и 8, но эти числа уже во второй строке есть. Тогда выполняется второй вариант $12 \to 6$, получается третья строка. По прежнему нельзя $9 \to 3$, но и $11 \to 7$ нельзя, так как тогда будет повторение 8. Следующей по разности парой будет $7 \to 4$, которая и выполнит операцию. Получим четвертую строку. $3 \to 9$ снова запрещено, но теперь можно $11 \to 6$, и получим пятую строку. И теперь ни одной операции не получится произвести, то есть эта строка и будет ответом.

Пример программы-решения

Ниже представлено решение на языке C++.

```
#include <bits/stdc++.h>

using namespace std;
typedef long long ll;
typedef pair<int, int> pii;
typedef long double ld;

vector<int> mask;

bool ok_turn(int a, int b){
    if(a > b)
        swap(a, b);

    if(mask[a+1] == 0 && mask[b-1] == 0 && a+2 != b)
        return 1;

    return 0;
}

int main(){
    ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);

    int n;
    cin >> n;
    vector<int> v(n);
    mask.resize(1001, 0);
    for(int i = 0; i < n; i++){
        cin >> v[i];
        mask[v[i]] = 1;
    }

    while(1){
    int big = -1, small = -1;

    for(int i = 1; i < n; i++)
        if(ok_turn(v[i-1], v[i])){
                int tbig = i-1;
                int tsmall = i;
                if(v[tbig] < v[tsmall])
                    swap(tbig, tsmall);
            if(big == -1){
                big = tbig;
```

```
                small = tsmall;
            }
            else{
                if(v[big] - v[small] < v[tbig] - v[tsmall] ||
                    (v[big] - v[small] == v[tbig] - v[tsmall] && v[tsmall] < v[small])){
                    big = tbig;
                    small = tsmall;
                }
            }
        }

        if(big == -1) break;
        mask[v[big]] = 0;
        v[big]--;
        mask[v[big]] = 1;
        mask[v[small]] = 0;
        v[small]++;
        mask[v[small]] = 1;
    }

    for(auto q : v) cout<<q<<' ';
    cout<<endl;
    return 0;
}
```

Задача III.1.1.3. Проектирование электросети (15 баллов)

Рассмотрим проект модели снабжения мегаполиса электроэнергией. Имеется n источников, генерирующих электроэнергию для нужд мегаполиса и уже запланированная структура линий электропередач, соединяющая эти источники с мегаполисом. Совокупно источники и мегаполис будем называть объектами. Для каждого источника известно, какую мощность он генерирует, кроме того известно, какие объекты будут соединены между собой передающими линиями. Известно, что вся сеть имеет древовидную структуру, то есть она связная и не имеет в своем составе циклов. Считаем, что мегаполис потребит всю мощность, которую получится доставить. Потери мощности при передаче считаем равными нулю.

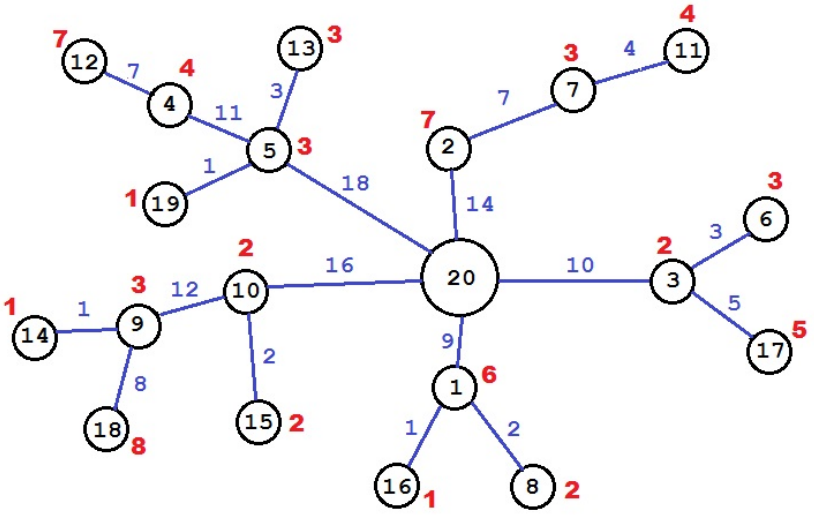
Требуется рассчитать пропускную способность каждой линии в проектируемой сети, то есть наибольшую мощность, которую можно передать по этой линии. При этом нужно указать минимально необходимую пропускную способность, такую, чтобы все имеющиеся генерируемые мощности были доставлены до мегаполиса.

Формат входных данных

В первой строке содержится число n — количество источников электроэнергии. Источники пронумерованы от 1 до n. Мегаполис имеет номер $n+1$. Далее в n следующих строках содержится по два числа a_i, b_j — обозначающих, что объекты номер a_i и b_j по проекту будут соединены линией. В последней строке содержатся n чисел gp_i через пробел, описывающих генерируемую мощность источника номер i. Все числа на входе в пределах от 1 до 1000. Гарантируется, что заданный граф является деревом, то есть связен и не содержит циклов.

Формат выходных данных

Для каждой линии в соответствующей ей отдельной строке указать необходимую минимальную пропускную способность. Порядок линий на выходе должен совпадать с порядком линий на входе.



Примеры

Пример №1

Стандартный ввод
19
12 4
19 5
5 13
4 5
5 20
2 20
20 10
7 2
10 15
10 9
18 9
14 9
1 20
1 8
16 1
20 3
3 6
17 3
7 11
6 7 2 4 3 3 3 2 3 2 4 7 3 1 2 1 5 8 1

Стандартный вывод
7 1 3 11 18 14 16 7 2 12 8 1 9 2 1 10 3 5 4

Пояснение к примеру

На рисунке приведена схема для первого примера. Красным обозначены генерируемые мощности, синим — необходимые минимальные пропускные способности линий. В частности, видно, что восточный блок источников 3, 6 и 17 генерирует суммарную мощность равную 10, а юго-западный подблок 9, 14, 18 требует для себя линии с пропускной способностью 12, и т. д.

Пример программы-решения

Ниже представлено решение на языке C++.

```cpp
#include <bits/stdc++.h>

using namespace std;
typedef long long ll;
typedef pair<int, int> pii;
typedef long double ld;

int n, a, b;
vector<vector<pii> > G;
vector<int> gp, ans;

void dfs(int a, int p){
    for(auto q : G[a])
        if(q.first != p){
            dfs(q.first, a);
            ans[q.second] = gp[q.first];
            gp[a] += gp[q.first];
        }
}

int main(){
```

```
    ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);

    cin >> n;
    n++;
    G.resize(n+2);
    gp.resize(n+1);
    ans.resize(n-1);
    for(int i = 0; i < n-1; i++){
        cin >> a >> b;
        G[a].push_back({b, i});
        G[b].push_back({a, i});
    }

    for(int i = 1; i < n; i++)
        cin >> gp[i];

    dfs(n, n);

    for(int i = 0; i < ans.size(); i++)
        cout << ans[i] <<endl;

    return 0;
}
```

Пример программы-решения

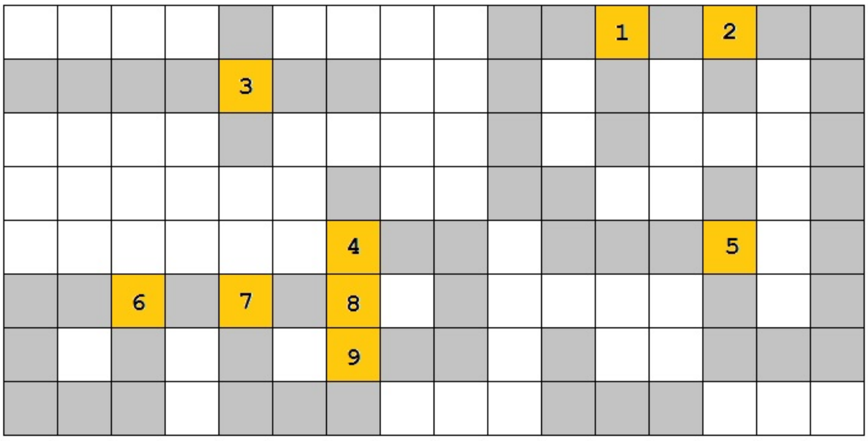
Ниже представлено решение на языке Python.

```
import sys
def dfs(u, par):
    global getId, a, dp, ans
    for elem in a[u]:
        if elem != par:
            dfs(elem, u)
            dp[u] += dp[elem]
            ans[getId[u][elem]] = dp[elem]
sys.setrecursionlimit(100000)
n = int(input())
n += 1
a = []
getId = []
ans = [0] * (n + 1)
dp = [0] * (n + 1)
for i in range(n + 1):
    a.append([])
    add = []
    for j in range(n + 1):
        add.append(-1)
    getId.append(add)
for i in range(n - 1):
    sd = input().split()
    u = int(sd[0])
    v = int(sd[1])
    a[u].append(v)
    a[v].append(u)
    getId[u][v] = i + 1
    getId[v][u] = i + 1
oo = list(map(int, input().split()))
for i in range(len(oo)):
```

```
    dp[i + 1] = oo[i]
dfs(n, -1)
for i in range(1, n):
    print(ans[i])
```

Задача III.1.1.4. Матрица кратчайших расстояний (25 баллов)

Рассмотрим уже известную модель дорожной сети, получаемую при анализе снимка местности. Изображение состоит только из точек двух типов. Точка, отмеченная 1, изображает дорожное полотно, все остальное изображается точками 0. При этом ширина дороги всегда ровно 1 точка. Нигде на изображении нет квадрата 2×2, состоящего из точек 1. Будем считать, что в точках, являющихся пересечением трех или четырех дорог находятся населенные пункты. Пронумеруем эти населенные пункты от 1 до K, сверху вниз и слева направо (см. пример, $K = 9$).



В данном случае нужно решить классическую задачу теории графов для не самого классического способа задания этих графов — при помощи упрощенного рисунка. А именно, нужно для каждой пары населенных пунктов определить кратчайшее расстояние между ними при движении по дорогам, или выяснить, что по дорогам попасть из одного в другой не получится. Будем считать, что перемещаться можно только из одной клетки помеченной 1 в соседнюю по стороне, так же помеченную 1, при этом длина такого перемещения равна 1.

Формат входных данных

Первая строка содержит два числа N и M через пробел — размер изображения. Далее в N строках содержится по M символов 0 или 1 без пробелов между ними, соответствующих изображению. Нигде на изображении нет квадрата 2×2, состоящего из точек 1. N и M находятся в пределах от 5 до 50. Гарантируется, что есть хотя бы один населенный пункт.

Формат выходных данных

Вывести K строк, по K чисел в каждой — матрицу кратчайших расстояний. В i-й строке на j-й позиции должно находиться кратчайшее расстояние между пунктами номер i и номер j. Обратите внимание на формат вывода: каждое расстояние требуется выводить в формате пять знаков, недостающие символы нужно заменять пробелами. Если из одного пункта попасть в другой нельзя, нужно выводить в таком случае -1.

Примеры

Пример №1

Стандартный ввод

```
8 16
0000100001111111
1111111001010101
0000100001010001
0000001001100101
0000001110111101
1111111010000101
1010101110100111
1110111000111000
```

Стандартный вывод

```
    0    2   -1   -1   10   -1   -1   -1   -1
    2    0   -1   -1   12   -1   -1   -1   -1
   -1   -1    0   -1   -1   -1   -1   -1   -1
   -1   -1   -1    0   -1    5    3    1    2
   10   12   -1   -1    0   -1   -1   -1   -1
   -1   -1   -1    5   -1    0    2    4    5
   -1   -1   -1    3   -1    2    0    2    3
   -1   -1   -1    1   -1    4    2    0    1
   -1   -1   -1    2   -1    5    3    1    0
```

Пример программы-решения

Ниже представлено решение на языке C++.

```cpp
#include <bits/stdc++.h>

using namespace std;
typedef long long ll;
typedef pair<int, int> pii;
typedef long double ld;

int n, m;
vector<string> S;
vector<vector<int> > M, H;
int dx[4] = {0, -1, 0, 1};
int dy[4] = {-1, 0, 1, 0};

bool intr(int x, int y){
    return (x >= 0 && y >= 0 && x < n && y < m);
}

bool np(int x, int y){
    int r = 0;
    for(int i = 0; i < 4; i++){
        int nx = x + dx[i];
```

```
        int ny = y + dy[i];
        if(intr(nx, ny))
            r += S[nx][ny] - '0';
    }

    return (r >= 3 && S[x][y] == '1');
}

void dfs(int x, int y, int d, int p){
    if(M[x][y] == 0)
        M[x][y] = 1;
    for(int i = 0; i < 4; i++){
        int nx = x + dx[i];
        int ny = y + dy[i];

        if(intr(nx, ny) && S[nx][ny] == '1'){
            if(M[nx][ny] == 0)
                dfs(nx, ny, d+1, p);
            else
                if(M[nx][ny] < 0 && -M[nx][ny] != p){
                   int np = -M[nx][ny];
                    H[np][p] = min(H[np][p], d+1);
                    H[p][np] = min(H[p][np], d+1);
                }
        }
    }
}

int main(){
    ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);

    cin >> n >> m;
    S.resize(n);
    for(int i = 0; i < n; i++)
        cin >> S[i];

    M.resize(n, vector<int> (m, 0));

    int K = 0;
    for(int i = 0; i < n; i++)
        for(int j = 0; j < m; j++)
        if(np(i, j)){
            K--;
            M[i][j] = K;
        }

        H.resize(-K+1, vector<int>(-K+1, 1e9));
        for(int i = 1; i <= -K; i++)
            H[i][i] = 0;

    for(int i = 0; i < n; i++)
        for(int j = 0; j < m; j++)
            if(M[i][j] < 0)
                dfs(i, j, 0, -M[i][j]);


    for (int k = 1; k <= -K; k++)
        for (int i = 1; i <= -K; i++)
            for (int j = 1; j <= -K; j++)
                H[i][j] = min (H[i][j], H[i][k] + H[k][j]);
```

```

        for(int i = 1; i <= -K; i++)
            for(int j = 1; j <= -K; j++)
                        if(H[i][j] == 1e9)
                            H[i][j] = -1;

        for(int i = 1; i <= -K; i++){
            for(int j = 1; j <= -K; j++)
            cout<<setw(5)<<H[i][j];
            cout<<endl;
       }
    return 0;
}
```

Пример программы-решения

Ниже представлено решение на языке Python.

```python
import queue
dx = [-1, 0, 1, 0]
dy = [0, 1, 0, -1]
n, m = list(map(int, input().split()))
a = [['0'] * (m + 2) for i in range(n + 2)]
check = [[-1] * (m + 2) for i in range(n + 2)]
for i in range(1, n + 1):
    om = input()
    for j in range(m):
        a[i][j + 1] = om[j]
yk = 1
ff = []
for i in range(1, n + 1):
    for j in range(1, m + 1):
        if a[i][j] == '1':
            cc = ord(a[i - 1][j]) + ord(a[i + 1][j]) + ord(a[i][j - 1]) + ord(a[i][j + 1]) - ord('0') - ord('0') - ord('0') - ord('0')
            if cc >= 3:
                check[i][j] = yk
                ff.append([i, j])
                yk += 1
for k in range(len(ff)):
    Q = queue.Queue()
    Q.put(ff[k])
    dist = [[1000000000] * (m + 2) for i in range(n + 2)]
    dist[ff[k][0]][ff[k][1]] = 0
    while Q.empty() == False:
        u = Q.get()
        for i in range(4):
            if (u[0] + dy[i] >= 1 and u[0] + dy[i] <= n and u[1] + dx[i] >= 1 and u[1] + dx[i] <= m and a[u[0] + dy[i]][u[1] + dx[i]] == '1' and dist[u[0] + dy[i]][u[1] + dx[i]] > dist[u[0]][u[1]] + 1):
                dist[u[0] + dy[i]][u[1] + dx[i]] = dist[u[0]][u[1]] + 1
                Q.put([u[0] + dy[i], u[1] + dx[i]])
    for i in range(len(ff)):
        if dist[ff[i][0]][ff[i][1]] == 1000000000:
            st = repr(-1).rjust(5)
            print(st,end='',flush = True)
        else:
            st=repr(dist[ff[i][0]][ff[i][1]]).rjust(5)
```

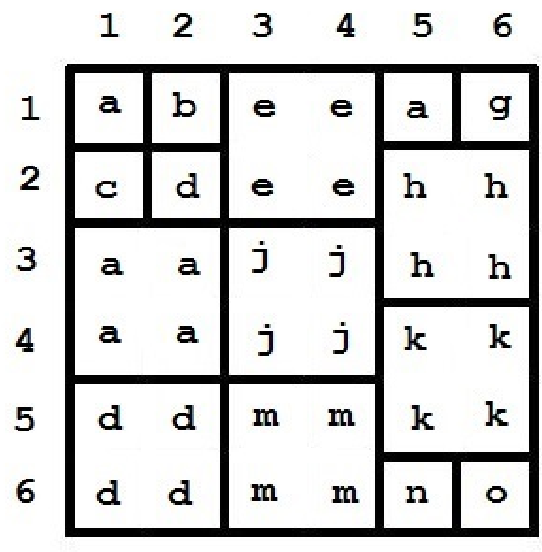
```
            print(st,end='',flush = True)
    print()
```

Задача III.1.1.5. Агломерация (33 баллов)

Агломерация — это «компактное скопление населенных пунктов, главным образом городов, местами срастающихся, объединенных в сложную многокомпонентную динамическую систему с интенсивными производственными, транспортными и культурными связями». Изначально отдельно образовавшиеся поселения со временем разрастаются и сливаются в одно многоуровневое образование — городскую агломерацию. При этом процесс слияния может иметь рекурсивную структуру, когда несколько агломераций сливаются в одну агломерацию более высокого уровня.

В этой задаче нужно построить обработку снимка городской агломерации, находящую ее уровень. Идея очевидна — если в агломерацию входит как часть агломерация уровня $n-1$ и нет частей-агломераций уровня n, то сама эта рассматриваемая агломерация имеет уровень n.

Но не все так просто — в данной задаче уточним понятие агломерации следующим образом: под агломерацией будем понимать любую квадратную область плоскости, целиком занятую отдельными поселениями, причем любое поселение, имеющееся в этом квадрате, не должно выходить за его пределы. Любой отдельный населенный пункт назовем агломерацией первого уровня. На снимке такой населенный пункт имеет вид квадрата, все ячейки которого обозначаются одним и тем же символом. Из низкоуровневых агломераций путем слияния получаются более высокоуровневые. Для заданного снимка квадратной области, целиком заполненной населенными кварталами, требуется определить его уровень.



Рассмотрим пример. Каждый единичный квадрат обозначен своим символом и является жилым кварталом. Если несколько рядом стоящих кварталов обозначены одним символом, они образуют населенный пункт, целиком занимающий квадратную область. Каждый такой населенный пункт образует агломерацию первого уровня. Всего таких на рисунке 15. Агломераций второго уровня на рисунке две, укажем их левый верхний и правый нижний углы: (1, 1) —- (2, 2) и (3, 1) —- (6, 4). Агломераций третьего уровня одна, ее координаты $(1, 1)$ — $(4, 4)$. Весь квадрат в примере является агломерацией уровня четыре.

Отдельно следует отметить, что разные населенные пункты могут быть обозначены одним символом, при условии, что они не имеют общую границу ненулевой длины.

Формат входных данных

В первой строке находится число n — размер стороны квадрата заданной агломерации ($1 \leqslant n \leqslant 100$). В следующих n строках содержится по n маленьких букв из множества a ... z без пробелов, описывающих населенные пункты в составе агломерации. Буквы одного вида, являющиеся соседними по стороне всегда образуют квадраты. Некоторые пары различных квадратов могут быть обозначены одинаковыми буквами, но при этом данные пары квадратов не имеют общих границ ненулевой длины.

Формат выходных данных

Вывести одно число — уровень представленной во входных данных агломерации.

Примеры

Пример №1

Стандартный ввод
6 abeeag cdeehh aajjhh aajjkk ddmmkk ddmmno

Стандартный вывод
4

Пример программы-решения

Ниже представлено решение на языке C++.

```
#include <bits/stdc++.h>

using namespace std;
typedef long long ll;
typedef pair<int, int> pii;
typedef long double ld;

int n;
vector<vector<int> > u, d, l ,r;

bool sq(int rx, int ry, int D){
```

```
    int lx = rx - D + 1;
    int ly = ry - D + 1;
    if(r[lx][ly] >= D && d[lx][ly] >= D && u[rx][ry] >= D && l[rx][ry] >= D)
        return 1;
    return 0;
}

int main(){
    ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);

    string s;
    cin >> n;
    vector<string> v(n+2, string(n+2, '#'));
    for(int i = 1; i <= n; i++){
        cin >> s;
        s = "#" + s + "#";
        v[i] = s;
    }

    u.resize(n+2, vector<int>(n+2, 0));
    for(int i = 1; i <= n; i++)
    for(int j = 1; j <= n; j++){
        u[i][j] = (v[i][j] != v[i][j+1]);
        u[i][j] += u[i-1][j] * (u[i][j] != 0);
    }

    l.resize(n+2, vector<int>(n+2, 0));
    for(int j = 1; j <= n; j++)
        for(int i = 1; i <= n; i++){
            l[i][j] = (v[i][j] != v[i+1][j]);
            l[i][j] += l[i][j-1] * (l[i][j] != 0);
        }

    d.resize(n+2, vector<int>(n+2, 0));
    for(int i = n; i >= 1; i--)
        for(int j = n; j >= 1; j--){
            d[i][j] = (v[i][j] != v[i][j-1]);
            d[i][j] += d[i+1][j] * (d[i][j] != 0);
        }

    r.resize(n+2, vector<int>(n+2, 0));
    for(int j = n; j >= 1; j--)
        for(int i = n; i >= 1; i--){
            r[i][j] = (v[i][j] != v[i-1][j]);
            r[i][j] += r[i][j+1] * (r[i][j] != 0);
        }

    vector<vector<int> > dp(n+2, vector<int>(n+2, 0));

    for(int u = 1; u <= n; u++)
        for(int i = n; i >= 1; i--)
            for(int j = n; j >= 1; j--)
                if(i - u + 1 >= 1 && j - u + 1 >= 1){
                    dp[i][j] = max({dp[i][j], dp[i][j-1], dp[i-1][j], dp[i-1][j-1]});
                    if(sq(i, j, u)) dp[i][j]++;
                }

    cout << dp[n][n]<<endl;

    return 0;
```

```
}
```

Пример программы-решения

Ниже представлено решение на языке Python.

```python
n = int(input())
a = [['#'] * (n + 2) for i in range(n + 2)]
for i in range(1, n + 1):
    sd = input()
    for j in range(1, n + 1):
        a[i][j] = sd[j - 1]
ans = []
k = [[0] * (n + 2) for i in range(n + 2)]
for i in range(n + 2):
    ans.append(k)
for sz in range(1, n + 1):
    for i in range(1, n - sz + 2):
        for j in range(1, n - sz + 2):
            y1 = i
            x1 = j
            x2 = j + sz - 1
            y2 = i + sz - 1
            xx1 = x1
            xx2 = x2
            ch = 1
            while x1 <= x2:
                if (a[y1 - 1][x1] == a[y1][x1] or a[y2][x1] == a[y2 + 1][x1]):
                    ch = 0
                    break
                x1 += 1

            while y1 <= y2:
                if (a[y1][xx1] == a[y1][xx1 - 1] or a[y1][xx2] == a[y1][xx2 + 1]):
                    ch = 0
                    break
                y1 += 1;
            ans[sz][i][j] =ch + max(ans[sz - 1][i][j], max(ans[sz - 1][i][j + 1],
            ↪   max(ans[sz - 1][i + 1][j], ans[sz - 1][i + 1][j + 1])))
print(ans[n][1][1])
```

Математика. 8–9 класс

Задача III.1.2.1. (10 баллов)

На одном из объектов атомной электростанции по регламенту круглосуточно должен находиться оператор. Дежурство одного оператора длится 8 часов, после чего его сменяет следующий сотрудник, причем брать две или три смены подряд в одни сутки категорически запрещается. Определите, сколько может быть различных вариантов расписаний дежурства на одни сутки, если на АЭС работают 8 операторов?

Решение

Поскольку в первые 8 часов есть возможность выбора из 8 операторов, во вторые 8 часов — только из 7, а третьи — только из 6, то количество вариантов расписаний дежурств выглядит следующим образом:

$$N = 8 \cdot 7 \cdot 6 = 336.$$

Ответ: 336.

Задача III.1.2.2. (15 баллов)

Известно, что шельфовые электростанции строят на участках моря с небольшой глубиной, при этом башни ветрогенераторов устанавливают на фундамент из свай, забитых на глубину до 30 метров. При установке опоры на море для ветрогенератора требуется произвести бурильные работы, причем бурение производилось на целое количество метров. За бурение первого метра бурильной компании требуется заплатить 218 тыс. руб. За каждый последующий метр цена должна снижаться на 5 тыс.руб. Кроме того, за вывоз грунта и сложные условия труда после работ бурильной компании было выплачено вознаграждение в размере 500 тыс. руб. На какую глубину (в метрах) было произведено бурение, если средняя стоимость бурения одного метра оказалась равной $188\frac{5}{22}$ тыс. руб.

Решение

Поскольку каждый последующий метр дешевле предыдущего на 5 тыс.руб., то в общем виде цена n-го мера является n-ым членом арифметической прогрессией: $P_n = 218 - 5 \cdot (n-1) = 223 - 5n$.

С одной стороны, по условию задачи средняя стоимость бурения одного метра составляет $188\frac{5}{22}$ тыс.руб. и, значит, за весь заказ на бурение n метров заплачено всего $188\frac{5}{22} \cdot n = \frac{4141}{22}n$ тыс. руб.. С другой стороны, за бурение было заплачено (из условия, что цена каждого метра является членом арифметической прогрессии) с учетом вознаграждения в 500 тыс.: $\frac{218+223-5n}{2} \cdot n + 500$ тыс. руб. Приравняем получившиеся выражения и, упростив уравнение, решим его:

$$\frac{218 + 223 - 5n}{2} \cdot n + 500 = \frac{4141}{22}n,$$

$$11n^2 - 142n - 2200 = 0,$$

$$\left[\begin{array}{l} n = -\frac{100}{11}, \\ n = 22, \end{array}\right.$$

Очевидно, что глубина не может быть отрицательной величиной, поэтому в ответ можно записать глубину 22 метра.

Ответ: 22.

Задача III.1.2.3. (20 баллов)

На электростанции ведется постоянный мониторинг пиковых и минимальных нагрузок на электросеть. В один из дней было замечено, что в первой половине суток переходы от среднесуточного значения нагрузки в 5 МВт до минимальных и обратно, а также от среднесуточного значения нагрузки до пиковых и обратно графически очень хорошо ложатся на две параболы (см. чертеж к задаче): $y_{\text{минимальная}}(\text{МВт}) = a_2t^2 + b_2t + c_2$ и $y_{\text{пиковая}}(\text{МВт}) = a_1t^2 + b_1t + c_1$, соответственно ($t$ - время в часах). Так, в ночное время при прохождении минимальных нагрузок в $2:00$, $2:30$ и $4:00$ нагрузка составляла 4.85, 4.1 и 3.65 МВт, соответственно; во время утреннего нарастания потребляемой мощности в в $8:00$, $8:30$ и $9:00$ нагрузка составляла 6, 9.75 и 12 МВт, соответственно. Найти разницы между моментами времени, когда возникают пиковая и минимальная нагрузки Δt, а также между максимальным и минимальным значениями потребляемой мощности ΔN. Ответ записать в минутах и МВт в формате «(Δt;ΔN)», в качестве разделителя целой и десятичной части в дробях использовать точку.

Рис. III.1.1: Чертеж к задаче

Решение

Составим две системы уравнений для минимальной и пиковой нагрузок, подставляя вместо t время измерений, а вместо y — потребляемую мощность чтобы определить в уравнениях значения параметров $a_1, b_1, c_1, a_2, b_2, c_2$:

$$\begin{cases} a_2 \cdot 2^2 + b_2 \cdot 2 + c_2 = 4,85, \\ a_2 \cdot 25^2 + b_2 \cdot 2,5 + c_2 = 4,1, \\ a_2 \cdot 4^2 + b_2 \cdot 4 + c_2 = 3,65. \end{cases} \Rightarrow \begin{cases} a_2 = 0,6, \\ b_2 = -4,2, \\ c_2 = 10,85. \end{cases}$$

$$\begin{cases} a_1 \cdot 8^2 + b_1 \cdot 8 + c_1 = 6, \\ a_1 \cdot 8,5^2 + b_1 \cdot 8,5 + c_1 = 9,75, \\ a_1 \cdot 9^2 + b_1 \cdot 9 + c_1 = 12. \end{cases} \Rightarrow \begin{cases} a_1 = -3, \\ b_1 = 57, \\ c_1 = -258. \end{cases}$$

Получив параметры уравнений, можно их переписать в виде:

$$\begin{cases} y_{\text{мин}}(\text{МВт}) = 0,6t^2 - 4,2t + 10,85 = 0,6(t-3,5)^2 + 3,5, \\ y_{\text{пик}}(\text{МВт}) = -3t^2 + 57t - 258, \end{cases} \Rightarrow$$

$$\begin{cases} y_{\text{мин}}(\text{МВт}) = 0,6(t-3,5)^2 + 3,5, \\ y_{\text{пик}}(\text{МВт}) = -3(t-9,5)^2 + 12,75. \end{cases}$$

Отсюда сразу можно получить информацию и о моменте возникновения минимальных и максимальных нагрузок на сеть (3:30 и 9:30, соответственно), и о величинах нагрузок в эти моменты (3,5 и 12,75 МВт). Таким образом, искомые величины Δt и ΔN равны 6 часов или 360 минут и 9,25 МВт. В ответ запишем $(360; 9,25)$.

Ответ: (360; 9,25).

Задача III.1.2.4. (20 баллов)

При выполнении частного заказа панель солнечной батареи изготовили в форме равностороннего треугольника. В качестве элемента крепления предусмотрели отверстие, расстояние от которого до двух вершин панели равно 2 м, а до третьей вершины — 1 м. Найти площадь в (м^2) поверхности панели солнечной батареи, ответ округлить до десятых.

Решение

Пусть сторона треугольника равна a м. Опустим перпендикуляр $MD \perp AB$. Поскольку $AM = BM$, точки C, M, D лежат на одной прямой — высоте CD. В ΔADC и ΔADM имеем $(1 + MD)^2 = a^2 - \frac{a^2}{4}$, $MD^2 = 4 - \frac{a^2}{4}$. Тогда $a^2 = 4(4 - MD^2)$ и получаем квадратное уравнение:

$$(1 + MD)^2 = 3(4 - MD^2), \text{ или } 4MD^2 + 2MD - 11 = 0,$$

откуда $MD = \frac{3\sqrt{5}-1}{4}$ (второй корень не удовлетворяет условию). Далее находим $a^2 = 4(4 - MD^2) = 16 - \frac{46-6\sqrt{5}}{4} = \frac{3\sqrt{5}+9}{2}$, и, следовательно:

$$S = \frac{a^2\sqrt{3}}{4} = \frac{9\sqrt{3} + 3\sqrt{15}}{8} = 3,4 \text{ м}^2.$$

Ответ: 3,4.

Задача III.1.2.5. (25 баллов)

На ветряной электростанции произвели плановую замену всех ветрогенераторов мощностью 15 МВт на меньшее количество семнадцатимегаваттных аналогов. При этом генерировать такой ветряной парк стал на 35 МВт больше. Какое количество МВт стал генерировать ветропарк, если до замены было больше 20, но меньше 30 ветрогенераторов?

Решение

Пусть было y ветрогенераторов, а стало — x. Тогда можно составить уравнение:

$$17x - 15y = 35.$$

Решим данное диофантово уравнение (поскольку числа, описывающие количество ветрогенераторов, должны быть целочисленными) с помощью цепных дробей:

$$\frac{17}{15} = 1 + \frac{1}{7 + \frac{1}{2}}.$$

Далее заменим в данной дроби последнюю дробь на $\frac{1}{1}$ и вычислим значение получившейся дроби: $1 + \dfrac{1}{7 + \frac{1}{1}} = \frac{9}{8}$.

Разница между исходной дробью и получившейся даст нам два частных решения данного диофантова уравнения:

$$\frac{15}{17} - \frac{9}{8} = \frac{15 \cdot 8 - 17 \cdot 9}{17 \cdot 8} = \frac{1}{17 \cdot 8}.$$

В двух последних выражениях видно, что $15 \cdot 8 - 17 \cdot 9 = 1$. Умножив на 35 левую и правую часть равенства получим два частных решения:

$$x = 8 \cdot 35 = 280, y = 9 \cdot 35 = 315.$$

Тогда решениями уравнения выглядят следующим образом:

$$\begin{cases} x = 280 + 15t, \\ y = 315 + 17t. \end{cases}$$

Учитывая, что $20 < x < 30$, найдем значение параметра t:

$$20 < 280 + 15t < 30,\ -17\frac{1}{3} < t < -16\frac{2}{3}.$$

Следовательно для $t = -17$ решениями уравнения будет пара значений:

$$\begin{cases} x = 25, \\ y = 26. \end{cases}$$

Мощность ветропарка стала равной $17 \cdot 25 = 425$ МВт.

Ответ: 425.

Математика. 10–11 класс

Задача III.1.3.1. (20 баллов)

На экзамене по технике безопасности в энергетической компании присутствуют 17 опытных сотрудников в должности старших операторов, 7 еще более опытных сотрудников в должности ведущих операторов и 6 молодых сотрудников, недавно принятых на должности младших операторов. Известно, что все ведущие операторы могут сдать экзамен только на «отлично», старшие операторы — равновероятно на «отлично» и «хорошо», а младшие операторы — с равной вероятностью на оценки «хорошо», «удовлетворительно» или «неудовлетворительно». Для сдачи экзамена комиссия вызывает сразу троих сотрудников. Найти вероятность того, что экзаменуемые получат оценки «отлично», «хорошо» и «удовлетворительно» (в любом порядке). В ответ записать дробь в формате «(числитель; знаменатель)».

Решение

Всего должны быть на экзамене $7+17+6 = 30$ операторов. Может быть несколько вариантов исходов, при которых данное условие может быть выполнено:

H_1 — вызваны один младший оператор, один старший и один ведущий оператор;

H_2 — вызваны один младший оператор и два старших;

H_3 — вызваны два младших оператора и один старший оператор;

H_4 — вызваны два младших оператора и один ведущий оператор.

Тогда вероятности можно записать следующим образом:

$$P(H_1) = 6 \cdot \frac{7}{30} \cdot \frac{17}{29} \cdot \frac{6}{28} = \frac{51}{290},$$

$$P(H_2) = 3 \cdot \frac{17}{30} \cdot \frac{16}{29} \cdot \frac{6}{28} = \frac{204}{1015},$$

$$P(H_3) = 3 \cdot \frac{17}{30} \cdot \frac{6}{29} \cdot \frac{5}{28} = \frac{51}{812},$$

$$P(H_4) = 3 \cdot \frac{7}{30} \cdot \frac{6}{29} \cdot \frac{5}{28} = \frac{3}{116}.$$

Тогда общая вероятность имеет вид:

$$P = P(H_1) \cdot 1 \cdot \frac{1}{2} \cdot \frac{1}{3} + P(H_2) \cdot 1 \cdot \frac{2}{4} \cdot \frac{1}{3} + P(H_3) \cdot 1 \cdot \frac{1}{2} \cdot \frac{2}{9} + P(H_4) \cdot 1 \cdot \frac{2}{9} = \frac{46}{609}.$$

Ответ: (46; 609).

Задача III.1.3.2. (20 баллов)

На ветряной электростанции произвели плановую замену всех ветрогенераторов мощностью 13 МВт на меньшее количество двадцатиодномегаваттных аналогов. При этом генерировать такой ветряной парк стал на 31 МВт меньше. Какое количество МВт стал генерировать ветропарк, если до замены было больше 20, но меньше 30 ветрогенераторов?

Решение

Пусть было x ветрогенераторов, а стало - y. Тогда можно составить уравнение: $13x - 21y = 11$.

Решим данное диофантово уравнение (поскольку числа, описывающие количество ветрогенераторов, должны быть целочиленными) с помощью цепных дробей:

$$\frac{13}{21} = 0 + \cfrac{1}{1 + \cfrac{1}{1 + \cfrac{1}{1 + \cfrac{1}{1 + \cfrac{1}{1 + \frac{1}{2}}}}}}.$$

Далее заменим в данной дроби последнюю дробь на $\frac{1}{1}$ и вычислим значение получившейся дроби:

$$0+\cfrac{1}{1+\cfrac{1}{1+\cfrac{1}{1+\cfrac{1}{1+\cfrac{1}{1+\frac{1}{1}}}}}}=\frac{8}{13}.$$

Разница между исходной дробью и получившейся даст нам два частных решения данного диофантова уравнения:

$$\frac{13}{21}-\frac{8}{13}=\frac{13\cdot 13-21\cdot 8}{13\cdot 21}=\frac{1}{13\cdot 21}.$$

В двух последних выражениях видно, что $13\cdot 13-21\cdot 8=1$. Умножив на 31 левую и правую часть равенства получим два частных решения: $x=8\cdot 31=403$, $y=8\cdot 31=248$. Тогда решениями уравнения выглядят следующим образом:

$$\begin{cases} x=403+21t,\\ y=248+13t. \end{cases}$$

Учитывая, что $20<x<30$, найдем значение параметра t:

$$20<403+21t<30,\ -18\frac{5}{21}<t<-17\frac{16}{21}.$$

Следовательно для $t=-18$ решениями уравнения будет пара значений:

$$\begin{cases} x=25,\\ y=14. \end{cases}$$

Мощность ветропарка стала равной $13\cdot 25=325$ МВт.

Ответ: 325.

Задача III.1.3.3. (20 баллов)

Решить уравнение:

$$\frac{1+3+...+(2n+1)}{\frac{1}{2}+\frac{1}{6}+...+\frac{1}{2070}}=2070.$$

Решение

Рассмотрим знаменатель дроби: $\frac{1}{2}+\frac{1}{6}+...+\frac{1}{2070}=\frac{1}{1\cdot 2}+\frac{1}{2\cdot 3}+...+\frac{1}{45\cdot 46}$.

Далее попытаемся найти закономерность в вычислении такой суммы:

1)$\frac{1}{1\cdot 2}+\frac{1}{2\cdot 3}=\frac{2}{3}$;

2)$\frac{1}{1\cdot 2}+\frac{1}{2\cdot 3}+\frac{1}{3\cdot 4}=\frac{3}{4}$;

3) $\frac{1}{1\cdot 2}+\frac{1}{2\cdot 3}+\frac{1}{3\cdot 4}+\frac{1}{4\cdot 5}=\frac{4}{5}$.

Получается, что для n слагаемых выражение будет выглядеть следующим образом:

$$\frac{1}{1\cdot 2}+\frac{1}{2\cdot 3}+...+\frac{1}{n\cdot(n+1)}=\frac{n}{n+1}.$$

Проверим это утверждение для $n+1$ слагаемых:

$$\frac{1}{1\cdot 2}+\frac{1}{2\cdot 3}+...+\frac{1}{n\cdot(n+1)}+\frac{1}{(n+1)\cdot(n+2)}=\frac{n}{n+1}+\frac{1}{(n+1)\cdot(n+2)}=$$

$$=\frac{n(n+2)+1}{(n+1)(n+2)}=\frac{n+1}{n+2}.$$

Таким образом:

$$\frac{1}{1\cdot 2}+\frac{1}{2\cdot 3}+...+\frac{1}{45\cdot 46}=\frac{45}{46}.$$

Тогда:

$$\frac{1+3+...+(2n+1)}{\frac{1}{2}+\frac{1}{6}+...+\frac{1}{2070}}=2070;$$

$$\frac{1+3+...+(2n+1)}{\frac{45}{46}}=45\cdot 46.$$

В числителе данной дроби находится сумма арифметической прогрессии, равная $1+3+...+(2n+1)=\frac{1+(2n-1)}{2}n=n^2$. Тогда окончательно:

$$\frac{1+3+...+(2n+1)}{\frac{45}{46}}=2070;\ n^2=45^2;\ n=45.$$

Ответ: 45.

Задача III.1.3.4. (20 баллов)

Известно, что шельфовые электростанции строят на участках моря с небольшой глубиной, при этом башни ветрогенераторов устанавливают на фундамент из свай. На одном из морских участков принято решение заливать сваи в виде усеченной правильной четырехугольной пирамиды с высотой, равной $\frac{8}{9}$ от высоты исходной пирамиды. Известно, что боковая грань имеет постоянную заданную площадь, а наклон к плоскости основания составляет α. При каком значении α объем сваи будет наибольшим?

Решение

Объем пирамиды $PABCD$ можно записать как $V=\frac{1}{3}S_{ABCD}\cdot PO$. Пусть $AD=a$; тогда $S_{ABCD}=a^2$, а из ΔPOK можно найти $PO=OK\operatorname{tg}\alpha=\frac{a}{2}\operatorname{tg}\alpha$. Значит,

$$V=\frac{1}{3}a^2\cdot\frac{a}{2}\operatorname{tg}\alpha=\frac{1}{6}a^3\operatorname{tg}\alpha. \qquad \text{(III.1.1)}$$

Далее, пусть $S_{PDC} = Q = \frac{1}{2}a \cdot PK$, где $PK = \frac{OK}{\cos\alpha} = \frac{a}{2\cos\alpha}$. Поэтому $Q == \frac{a^2}{4\cos\alpha}$, откуда:

$$a = \sqrt{4Q\cos\alpha}. \qquad \text{(III.1.2)}$$

Подставив (III.1.2) в (III.1.1), получим:

$$V = V(\alpha) = \frac{1}{6}(\sqrt{4Q\cos\alpha})^3 \operatorname{tg}\alpha = \frac{4Q}{3}\sin\alpha\sqrt{Q\cos\alpha},$$

где $0 < \alpha < \frac{\pi}{2}$ по смыслу задачи. Если $\alpha \to 0$ и $\alpha \to \frac{\pi}{2}$, то $V(\alpha) \to 0$, причем $V(\alpha) > 0$ на $0 < \alpha < \frac{\pi}{2}$. Найдем производную объема и точки экстремума на заданном промежутке, приравняв производную к нулю:

$$V'(\alpha) = \frac{4}{3}Q^2\frac{2\cos^2\alpha - \sin^2\alpha}{2\sqrt{Q\cos\alpha}} = 0;$$

$$2\cos^2\alpha - \sin^2\alpha = 0;\ \operatorname{tg}^2\alpha = 2.$$

Получается, что $\operatorname{tg}\alpha = \pm\sqrt{2}$. Очевидно, что условию задачи удовлетворяет только значение $\operatorname{arctg}\sqrt{2}$.

Ответ: $\operatorname{arctg}\sqrt{2}$.

Задача III.1.3.5. (20 баллов)

Найти все пары чисел $(x; y)$, для каждой их которых выполняется равенство

$$12\sqrt{6y - y^2 - 5} \cdot \cos^2(\frac{3y - 5x}{2}) = 17 + 8\cos(3y - 5x) - 4\sin^2(3y - 5x).$$

Решение

Используя формулы: $\cos^2\frac{\alpha}{2} = \frac{1+\cos\alpha}{2}$ и $\sin^2\alpha = 1 - \cos^2\alpha$, получим:

$$6\sqrt{6y - y^2 - 5} \cdot (1 + \cos(3y - 5x)) = 17 + 8\cos(3y - 5x) - 4(1 - \cos^2(3y - 5x))$$

или

$$4\cos^2(3y - 5x) + 2(4 - 3\sqrt{6y - y^2 - 5})\cos(3y - 5x) + 13 - 6\sqrt{6y - y^2 - 5} = 0.$$

Рассматривая последнее уравнение как квадратное относительно $\cos(3y - 5x)$, получим $D = -9(y - 3)^2 \geqslant 0 \Rightarrow y = 3$, тогда при $y = 3$ $\cos(9 - 5x) = \frac{1}{2}$, получим:

$$x = \pm\frac{\pi}{15} + \frac{9}{5} + \frac{2\pi k}{5}.$$

Ответ: $(\pm\frac{\pi}{15} + \frac{9}{5} + \frac{2\pi k}{5}; 3), k \in Z.$