

## ЛАБОРАТОРНЫЙ ПРАКТИКУМ

### Лабораторная работа № 1. Вычисление времени выполнения программ

Найдите, используя  $O$ -символику, время выполнения алгоритма, а также дайте оценку временной эффективности в наихудшем случае. Используйте правила суммы и произведения. Получите  $T(n)$ .

#### Вариант 1.

```
import java.util.Scanner;

public class Main {
    static int n = 0, m = 0;
    static char[] s;
    static boolean f;
    public static void main(String[] args) throws
                                                Exception {

        Scanner sc = new Scanner(System.in);
        System.out.print("Введите N: ");
        n = sc.nextInt();
        System.out.print("Введите M <=N: ");
        m = sc.nextInt();

        String temp = "";
        for(int i = 49; i <= 48 + m; i++) {
            temp += (char) i;
        }
        s = temp.toCharArray();
        do {
            System.out.println(s);
            sochet();
        } while(!f);
    }

    static void sochet() {
        int i = poisk();
        if(i != -1) {
            s[i] = succ(s[i]);
            for(int j = i + 1; j < m ; j++) {
                s[j] = succ(s[j - 1]);
            }
        } else
            f = true;
    }
}
```

```

static int poisk() {
    for(int i = m - 1; i >= 0; i--)
        if (s[i] != (char) (n - m + i + 49))
            return i;
    return -1;
}

static char succ(char c) {
    return (char) ((int) c + 1);
}
}

```

## Вариант 2.

```

import java.util.Scanner;

public class Main {
    final static int MAX_M = 100;
    static int a[] = new int[MAX_M];
    static int m, n;
    public static void main(String[] args) throws Exception {
        Scanner sc = new Scanner(System.in);
        System.out.print("m = ");
        m = sc.nextInt();
        System.out.print("n (n <= m) = ");
        n = sc.nextInt();
        if (n <= m) {
            initMas();
            generator();
        }
    }

    static void initMas() {
        for(int i = 0; i < m; i++)
            a[i] = i + 1;
    }

    static void displayMas() {
        for(int i = 0; i < n; i++)
            System.out.print(a[i] + " ");

        System.out.println();
    }

    static void generator() {
        if(m == n)
            displayMas();
        else {
            int k = n;

```

```

        while(k >= 0) {
            displayMas();
            if(a[n - 1] == m)
                k--;
            else
                k = n - 1;

            if(k >= 0)
                for (int i = n - 1; i >= k; i--)
                    a[i] = a[k] + i - k + 1;
        }
    }
}

```

### Вариант 3.

```

import java.util.HashSet;
import java.util.Scanner;
import java.util.Set;

public class Main {
    static {
        final int MAX = 100;

        int i, n, j, bl;
        boolean vyh;
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Введите n = ");
        int n = sc.nextInt();
        char[] c = new char[n];

        System.out.println("Задайте символьную таблицу:");
        String temp = sc.nextLine();
        c = temp.split(" ");

        System.out.println("Введите слово:");
        String s = sc.nextLine();

        if(s.length() > n) {
            System.out.println("No");
            return;
        }

        int bl = 0;
        Set<Integer> m = new HashSet<Integer>();
    }
}

```

```

boolean vyh;

for (int i = 0; i < s.length(); i++) {
    vyh = true;
    int j = 0;
    while (vyh && j < n) {
        if(s.charAt(i) == c[j] && !m.contains(j)){
            vyh = false;
            m.add(j);
        }
        j++;
    }
    if(vyh) {
        System.out.println("No");
        return;
    }
}
System.out.println("Слово составить можно");
}
}

```

#### **Вариант 4.**

```

import java.util.HashSet;
import java.util.Scanner;
import java.util.Set;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Введите s");
        String s = sc.nextLine();
        String t = "";
        Set<Integer> b = new HashSet<>();
        int n = s.length();

        for (int i = 0; i < n; i++)
            for (int j = 0; j < n; j++)
                if (i != j && !b.contains(i) &&
                    !b.contains(j) &&
                    (s.charAt(i) == s.charAt(j))) {
                    b.add(i);
                    b.add(j);
                    t = s.charAt(i) + t + s.charAt(i);
                }

        if(n % 2 == 0)
            if (t.length() < n)
                System.out.println("No");
            else
                System.out.println(t);
    }
}

```

```

else {
    if(t.length() != n - 1) {
        System.out.println("No");
        return;
    } else {
        int j = 0;
        for (int i = 0; i < n; i++)
            if(!b.contains(i))
                j = i;

        for (int i = 0; i < (t.length() / 2); i++)
            System.out.print(t.charAt(i));

        System.out.print(s.charAt(j));
        for (int i = t.length() / 2;
              i < t.length(); i++)
            System.out.print(t.charAt(i));
    }
}
}
}

```

#### **Вариант 5.**

```

public class Main {
    public static void main(String[] args) {
        boolean p = true;
        long x1 = 12, x2, y1 = 0, y2 = 0;
        while (p) {
            x2 = 4;
            while (x2 != x1) {
                if ((x1*x1*x1 - x2*x2*x2) % 1000 != 0 &&
                    (x1*x1*x1 - x2*x2*x2) % 91 == 0) {
                    p = false;
                    y1 = x1;
                    y2 = x2;
                }
                x2 += 4;
            }
            x1 += 4;
        }
        System.out.println(y1 + " " + y2);
    }
}

```

#### **Вариант 6.**

```

public class Main {
    public static void main(String[] args) {
        int i = 1;
        boolean vyh = true;
    }
}

```

```

double s1, s2 = 0.0, h;
while (vyh) {
    i++;
    h = 1.0 / i;
    s1 = 0.5 * h * (1 + s(i, h) + 0.5);
    h *= 0.5;
    s2 = 0.5 * h * (1 + s(2 * i, h) + 0.5);
    if(Math.abs(s1 - s2) < 0.001)
        vyh = false;

    System.out.println(i);
}
System.out.println("_____");
System.out.println("N = " + i + " s = " + s2);
}

static double s(int k, double h) {
    double s1 = 0.0;
    for (int i = 1; i < k; i++)
        s1 += f(i * h);

    return 2 * s1;
}

static double f(double k) {
    return 1.0 / (1 + k * k * k);
}
}

```

### **Вариант 7.**

```

import java.util.HashSet;
import java.util.Scanner;
import java.util.Set;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Введите строку ");
        String s = sc.nextLine();
        System.out.println("Введите p ");
        double p = sc.nextDouble();

        int n = 90;
        JustClass a[] = new JustClass[n];
        Set<Character> w = new HashSet<Character>();
        int a1 = -1;
    }
}

```

```

        for (int i = 0; i < n; i++) {
            a[i] = new JustClass();
            a[i].z = 0;
        }

        for (int i = 0; i < s.length(); i++) {
            if(!w.contains(s.charAt(i))) {
                w.add(s.charAt(i));
                al++;
                a[al].x = s.charAt(i);
                a[al].z = 1;
            } else {
                for (int j = 0; j < al; j++)
                    if(a[j].x == s.charAt(i))
                        a[j].z++;
            }
        }

        System.out.println("Удовлетворяющие условию
                               СИМВОЛЫ (если они есть) ");
        for (int i = 0; i <= al; i++) {
            if(100.0 * a[i].z / al >= p) {
                if(a[i].x == ' ')
                    System.out.print(" Текст 2 ");
                else
                    System.out.print(a[i].x);
            }
        }
    }

    static class JustClass {
        char x;
        int z;
    }
}

```

### **Вариант 8.**

```

import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Введите размеры таблицы");
        int x = sc.nextInt(), y = sc.nextInt();
        int[][] a = vvod(x,y);
        System.out.println();
        double s = 1;
        System.out.println("Все средние значения:");
    }
}

```

```

        for (int i = 1; i < x - 1; i++) {
            for (int j = 1; j < y - 1; j++) {
                double sr1 = a[i - 1][j - 1] *
                    a[i + 1][j - 1] * a[i + 1][j + 1] *
                    a[i - 1][j + 1];
                double sr = Math.exp(Math.log(sr1) * 0.25);
                System.out.println(sr);
                if(a[i][j] < sr)
                    s *= a[i][j];
            }
        }
        System.out.println("Ответ: произведение равно "
                                + s);
    }

    static int[][] vvod(int x, int y) {
        int[][] a = new int[x][y];
        for (int i = 0; i < x; i++) {
            for (int j = 0; j < y; j++)
                a[i][j] = (int) (Math.random() * 5);
        }

        System.out.println("Массив A -> ");
        for (int i = 0; i < x; i++) {
            for (int j = 0; j < y; j++)
                System.out.print(a[i][j] + " ");

            System.out.println();
        }
        return a;
    }
}

```

### **Вариант 9.**

```

import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Введите n = ");
        int n = sc.nextInt();
        int[] a = vvod(n);

        for (int i = 1; i < n; i++) {
            if(a[i] < a[0]) {
                int t = a[i];
                for (int p = i - 1; p >= 0 ; p--)
                    a[p + 1] = a[p];
            }
        }
    }
}

```



```

        a[0] = t;
    } else {
        for (int p = 0; p < i - 2; p++) {
            if(a[i] >= a[p] && a[i] <= a[p + 1]) {
                int t = a[i];
                for(int j = i - 1; j >= p+1; j--)
                    a[j + 1] = a[j];

                a[p + 1] = t;
            }
        }
    }
}

System.out.println();
for (int i = 0; i < n; i++)
    System.out.print(a[i] + " ");
}

static int[] vvod(int n) {
    int[] a = new int[n];
    for (int i = 0; i < n; i++)
        a[i] = (int) (Math.random() * 50);

    for (int i = 0; i < n; i++)
        System.out.print(a[i] + " ");

    return a;
}
}

```

### **Вариант 10.**

```

import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Введите n = ");
        int n = sc.nextInt();
        int[] a = vvod(n);
        System.out.println();
        int max = a[0], min = a[0];
        for (int i = 1; i < n; i++) {
            if(a[i] > max)
                max = a[i];

            if(a[i] < min)
                min = a[i];
        }
    }
}

```

```

double s = 0.0;
for (int i = 0; i < n; i++)
    s += 1.0 / a[i];

s = n / s;
System.out.print("max = " + max);
System.out.print("; min = " + min);
System.out.println("; s = " + s);

if(Math.abs(s - min) < Math.abs(max - s))
    System.out.println("min = " + min +
        " меньше удален от среднего гармонического");

if(Math.abs(s - min) > Math.abs(max - s)) {
    System.out.println("max = " + max +
        " меньше удален от среднего гармонического");

if(Math.abs(s - min) == Math.abs(max - s)){
    System.out.println("max и min одинаково
        удалены от среднего гармонического");
}

static int[] vvod(int n) {
    int[] a = new int[n];
    for (int i = 0; i < n; i++) {
        a[i] = 0;
        while (a[i] == 0)
            a[i] = (int) (Math.random()*100 - 80);
    }

    System.out.println("Массив A -> ");
    for (int i = 0; i < n; i++)
        System.out.print(a[i] + " ");

    return a;
}
}

```

### Вариант 11.

```

public class Main {
    public static void main(String[] args) {
        boolean found = false;
        double a1 = -6, a2 = -5, b1 = -5, b2 = -4;
        double a = 0, b = 0;
        while (!found) {
            if(f(a1) * f(a2) < 0) {
                found = true;
                a = a1;
                b = a2;
            }
        }
    }
}

```

```

        if(f(b1) * f(b2) < 0) {
            found = true;
            a = b1;
            b = b2;
        }

        a1--; a2--;
        b1++; b2++;
    }

    if(f(a) * f(b) > 0) {
        System.out.println("Корней на участке нет");
        return;
    }
    if(f(a) == 0) {
        System.out.println("Корень = " + a);
        return;
    }
    if(f(b) == 0) {
        System.out.println("Корень = " + b);
        return;
    }
    double c = (a + b) / 2;
    while (Math.abs(f(c)) > 0.0001) {
        if(f(c) * f(b) > 0)
            b = c;
        else
            a = c;

        c = (a + b) / 2;
    }
    System.out.println(c);
}

static double f(double x) {
    return 0.8896*x*x*x - 2.813*x*x - 3.6929*x + 11.2;
}
}

```

## Лабораторная работа № 2. Анализ рекурсивных программ. Решение рекуррентных соотношений

Решите, согласно вариантам, следующие рекуррентные уравнения:

- а) методом подстановки;
- б) используя общее решение классов рекуррентных уравнений.

1.  $T(1) = 3; T(n) = 2T(3n / 4) + n;$
2.  $T(1) = 2n; T(n) = 2T(2n / 3) + n^2;$

3.  $T(1) = 1; T(n) = 9T(n/9) + n^3;$
4.  $T(1) = 1; T(n) = 4T(n/4) + n;$
5.  $T(1) = 1; T(n) = 2T(n/3) + n^2;$
6.  $T(1) = 1; T(n) = 6T(n/3) + n^3;$
7.  $T(1) = 3; T(n) = 2T(n/2) + (\log_2 n)^2;$
8.  $T(1) = 51; T(n) = T(3n/8) + n \cdot \log n;$
9.  $T(1) = 27; T(n) = 2T(5n/8) + n^2;$
10.  $T(1) = 15; T(n) = 8T(n/4) + n^3;$
11.  $T(1) = n/2; T(n) = 15T(3n/5) + 1;$
12.  $T(1) = 6n; T(n) = 6T(5n/6) + n - 1.$

**Решение задачи методом подстановки.** Пусть дано рекуррентное соотношение

$$T(1) = t_1(n), \quad T(n) = aT(n/b) + f(n), \quad (1)$$

где  $a \geq 1, b > 1, t_1(n)$  – число или некоторая заданная функция, зависящая от  $n$ ;  $f(n)$  – также заданная функция.

Найдем решение уравнения (1)

$$\begin{aligned} T(n) &= aT(n/b) + f(n) = \{T(n/b) = aT(n/b^2) + f(n/b)\} = \\ &= a(aT(n/b^2) + f(n/b)) + f(n) = a^2T(n/b^2) + af(n/b) + f(n) = \\ &= \{T(n/b^2) = aT(n/b^3) + f(n/b^2)\} = a^2(aT(n/b^3) + f(n/b^2)) + af(n/b) + f(n) = \\ &= a^3T(n/b^3) + a^2f(n/b^2) + af(n/b) + f(n) = \dots = \\ &= a^kT(n/b^k) + a^{k-1}f(n/b^{k-1}) + a^{k-2}f(n/b^{k-2}) + \dots + af(n/b) + f(n). \end{aligned}$$

Примем  $n/b^k = 1$ , следовательно,  $b^k = n$  или  $k = \log_b n$ . Тогда

$$T(n) = a^{\log_b n} T(1) + \sum_{j=0}^{k-1} a^j f(n/b^j). \quad (2)$$

Далее

$$\begin{aligned} a^{\log_b n} T(1) &= a^{\log_b n} t_1(n) = \left\{ \log_b n = \frac{\log_a n}{\log_a b} = \log_a n^{\frac{1}{\log_a b}} = \log_a n^{\log_b a} \right\} = \\ &= a^{\log_a n^{\log_b a}} t_1(n) = n^{\log_b a} t_1(n). \end{aligned}$$

Вычислим значение суммы в формуле (2) для некоторых частных случаев.

1.  $f(n) = n^m$ ,  $m \geq 0$ . Тогда

$$\sum_{j=0}^{k-1} a^j f(n/b^j) = \sum_{j=0}^{k-1} a^j \left(\frac{n}{b^j}\right)^m = n^m \sum_{j=0}^{k-1} \left(\frac{a}{b^m}\right)^j = S_k.$$

Получили геометрическую прогрессию. Первый член  $b_1 = n^m$ , знаменатель  $q = \frac{a}{b^m}$ , число членов равно  $k = \log_b n$ . Вычислим ее сумму по формуле

$$S_k = b_1 \frac{q^k - 1}{q - 1} = n^m \frac{(a/b^m)^k - 1}{a/b^m - 1} = \frac{n^m}{a/b^m - 1} \left( (a/b^m)^{\log_b n} - 1 \right). \quad (3)$$

Вычислим далее

$$\begin{aligned} \left(a/b^m\right)^{\log_b n} &= \frac{a^{\log_b n}}{b^{m \log_b n}} = \frac{a^{\log_b n}}{b^{\log_b n^m}} = \frac{a^{\frac{\log_a n}{\log_a b}}}{n^m} = \\ &= \frac{a^{\log_a n^{1/\log_a b}}}{n^m} = \frac{n^{\frac{1}{\log_a b}}}{n^m} = \frac{n^{\log_b a}}{n^m} = n^{\log_b a - m}. \end{aligned}$$

Подставляя данное выражение в (3), получим

$$S_k = \frac{n^m}{a/b^m - 1} \left( n^{\log_b a - m} - 1 \right) = \frac{n^{\log_b a} - n^m}{a/b^m - 1}.$$

Таким образом,

$$T(n) = n^{\log_b a} t_1(n) + \frac{n^{\log_b a} - n^m}{a/b^m - 1}.$$

2.  $f(n) = \log_c n$ ,  $c \geq 0$ . Тогда

$$\begin{aligned} \sum_{j=0}^{k-1} a^j f(n/b^j) &= \sum_{j=0}^{k-1} a^j \log_c (n/b^j) = \sum_{j=0}^{k-1} a^j (\log_c n - \log_c b^j) = \\ &= \sum_{j=0}^{k-1} a^j (\log_c n - j \log_c b) = \log_c n \sum_{j=0}^{k-1} a^j - \log_c b \sum_{j=1}^{k-1} j a^j = \tilde{S}_k + \tilde{\tilde{S}}_{k-1}. \end{aligned}$$

Вычислим значение первой суммы в последнем выражении

$$\begin{aligned}\tilde{S}_k &= \log_c n \sum_{j=0}^{k-1} a^j = \{b_1 = \log_c n; q = a\} = b_1 \frac{q^k - 1}{q - 1} = \log_c n \frac{a^k - 1}{a - 1} = \\ &= \log_c n \frac{a^{\log_b n} - 1}{a - 1} = \log_c n \frac{a^{\log_a n^{\log_b a}} - 1}{a - 1} = \log_c n \frac{n^{\log_b a} - 1}{a - 1} = \frac{(n^{\log_b a} - 1) \log_b n}{(a - 1) \log_b c}.\end{aligned}$$

Далее

$$\begin{aligned}\tilde{\tilde{S}}_{k-1} &= -\log_c b \sum_{j=1}^{k-1} j a^j = -a \log_c b \sum_{j=1}^{k-1} j a^{j-1} = -a \log_c b \sum_{j=1}^{k-1} (a^j)' = \\ &= -a \log_c b \left( \sum_{j=1}^{k-1} a^j \right)' = (-a \log_c b) \hat{S}'_{k-1}(a).\end{aligned}$$

Вычислим значение суммы  $\hat{S}_{k-1} = \sum_{j=1}^{k-1} a^j$ .

$$\hat{S}_{k-1} = \{b_1 = a; q = a\} = b_1 \frac{q^{k-1} - 1}{q - 1} = a \frac{a^{k-1} - 1}{a - 1} = \frac{a^k - a}{a - 1}.$$

Тогда

$$\begin{aligned}\tilde{\tilde{S}}_{k-1} &= -a \log_c b \left( \frac{a^k - a}{a - 1} \right)' = -a \log_c b \frac{(a^k - a)'(a - 1) - (a - 1)'(a^k - a)}{(a - 1)^2} = \\ &= -a \log_c b \frac{(ka^{k-1} - 1)(a - 1) - (a^k - a)}{(a - 1)^2} = -\frac{a \log_c b}{(a - 1)^2} ((k - 1)a^k - ka^{k-1} + 1) = \\ &= -\frac{a \log_c b}{(a - 1)^2} ((k - 1)a^{k-1}a - ka^{k-1} + 1) = -\frac{a \log_c b}{(a - 1)^2} (ka^{k-1}a - a^k - ka^{k-1} + 1) = \\ &= -\frac{a \log_c b}{(a - 1)^2} (ka^{k-1}(a - 1) - a^k + 1) = -\log_c b \left( \frac{1}{a - 1} ka^k - \frac{a}{(a - 1)^2} a^k + \frac{a}{(a - 1)^2} \right) = \\ &= -\log_c b \left( \frac{1}{a - 1} \log_b n a^{\log_b n} - \frac{a}{(a - 1)^2} a^{\log_b n} + \frac{a}{(a - 1)^2} \right) =\end{aligned}$$

$$\begin{aligned}
&= -\log_c b \left( \frac{1}{a-1} \log_b n a^{\log_a n^{\log_b a}} - \frac{a}{(a-1)^2} a^{\log_a n^{\log_b a}} + \frac{a}{(a-1)^2} \right) = \\
&= -\log_c b \left( \frac{1}{a-1} n^{\log_b a} \log_b n - \frac{a}{(a-1)^2} n^{\log_b a} + \frac{a}{(a-1)^2} \right).
\end{aligned}$$

Тогда

$$\begin{aligned}
&\sum_{j=0}^{k-1} a^j f(n/b^j) = \tilde{S}_k + \tilde{S}_{k-1} = \frac{(n^{\log_b a} - 1) \log_b n}{(a-1) \log_b c} - \\
&- \log_c b \left( \frac{1}{a-1} n^{\log_b a} \log_b n - \frac{a}{(a-1)^2} n^{\log_b a} + \frac{a}{(a-1)^2} \right) = \\
&= \left( \frac{1}{(a-1) \log_b c} - \frac{\log_c b}{a-1} \right) n^{\log_b a} \log_b n - \frac{\log_b n}{(a-1) \log_b c} + \frac{a \log_c b}{(a-1)^2} n^{\log_b a} - \\
&- \frac{a \log_c b}{(a-1)^2} = \frac{a \log_c b}{(a-1)^2} n^{\log_b a} - \frac{\log_c b \log_b n}{a-1} - \frac{a \log_c b}{(a-1)^2}.
\end{aligned}$$

Таким образом,

$$T(n) = \left( t_1(n) + \frac{a \log_c b}{(a-1)^2} \right) n^{\log_b a} - \frac{\log_c b \log_b n}{a-1} - \frac{a \log_c b}{(a-1)^2}.$$

Далее, подставляя конкретные значения параметров, учитывая асимптотическое поведение слагаемых, переходим к  $O$ -символике.

### Лабораторная работа № 3. АД Множество

Реализуйте АД Множество и на его основе решите задачи по вариантам.

1. Дан текст из строчных латинских букв, за которым следует точка. Напечатайте:

- первые вхождения букв в текст, сохраняя их взаимный исходный порядок;
- все буквы, входящие в текст не менее двух раз;
- все буквы, входящие в текст по одному разу.

Дана непустая последовательность слов из строчных русских букв; между соседними словами – запятая, за последним словом – точка. Напечатайте в алфавитном порядке:

2. Все гласные буквы, которые входят в каждое слово; все согласные буквы, которые не входят ни в одно слово.

3. Все звонкие согласные буквы, которые входят хотя бы в одно слово; все глухие согласные буквы, которые не входят хотя бы в одно слово.

4. Все согласные буквы, которые входят только в одно слово; все глухие согласные буквы, которые не входят только в одно слово.

5. Все звонкие согласные буквы, которые входят более чем в одно слово; все гласные буквы, которые не входят более чем в одно слово.

6. Все звонкие согласные буквы, которые входят в каждое нечетное слово и не входят ни в одно четное слово; все глухие согласные буквы, которые входят в каждое нечетное слово и не входят хотя бы в одно четное слово.

7. Подсчитайте общее количество цифр и знаков '+', '-', и '\*', входящих в строку  $s$ .

8. Подсчитайте количество различных (значащих) цифр в десятичной записи натурального числа  $n$  и напечатать в возрастающем порядке все цифры, не входящие в десятичную запись натурального числа  $n$ .

9. Вычислите сумму тех элементов матрицы  $A$ , номера строк и столбцов которых принадлежат соответственно непустым множествам  $S_1$  и  $S_2$ .

#### **Задачи повышенной сложности.**

10. Дан текст, состоящий из русских слов. Выполните разделение каждого его слова на части для переноса.

Далее правила переноса русских слов.

А. Две идущие подряд гласные можно разделить, если первой из них предшествует согласная, а за второй идет хотя бы одна буква (буква «й» при этом рассматривается вместе с предшествующей гласной как единое целое).

Б. Две идущие подряд согласные можно разделить, если первой из них предшествует гласная, а в той части слова, которая идет за второй согласной, имеется хотя бы одна гласная (буквы «ъ», «ь» вместе с предшествующей согласной рассматриваются как единое целое).

В. Если не удастся применить пункты А, Б, то следует попытаться разбить слова так, чтобы первая часть содержала более чем одну букву и оканчивалась на гласную, а вторая содержала хотя бы одну гласную. Вероятность правильного разбиения увеличивается, если предварительно воспользоваться хотя бы неполным списком приставок, содержащих гласные, и попытаться, прежде всего, выделить из слова такую приставку.

11. Напишите программу раздачи карт при игре в «дурака», количество игроков задается с клавиатуры.

12. Для произвольного символьного множества сгенерируйте все подмножества.



## Лабораторная работа № 4. АТД Список

Реализуйте АТД Список и на его основе решите задачи по вариантам.

1. Сформируйте список строк и а) сохраните его в текстовом файле; б) сохраните его в обратном порядке в текстовом файле. Используйте рекурсию.
2. Сформируйте список строк из текстового файла.
3. Напишите функцию, которая вычисляет среднее арифметическое элементов непустого списка.
4. Напишите процедуру присоединения списка  $L2$  к списку  $L1$ .
5. Напишите функцию, которая создает список  $L2$ , являющийся копией списка  $L1$ , начинающегося с данного узла.
6. Напишите функцию, которая подсчитывает количество вхождений ключа в списке.
7. Напишите функцию, которая удаляет из списка все вхождения ключа.
8. Многочлен  $P_n(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$  задан своими коэффициентами, которые хранятся в форме списка. Напишите функции:
  - $Equal(p, q)$ , проверяющую на равенство многочлены  $p$  и  $q$ ;
  - $Sum(p, q, r)$ , которая строит многочлен  $r = p + q$ .
9. Вычислите значение многочлена  $P_n(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$  в целочисленной точке  $x$ . Коэффициенты вводятся с клавиатуры и динамически размещаются в памяти.
10. Сформируйте список целых чисел и упорядочьте их по неубыванию.
11. Сформируйте список целых чисел и удалите из него все четные.
12. Сформируйте список вещественных чисел и вычислите сумму.

## Лабораторная работа № 5. Методы сортировки данных

**Задание 1.** Реализуйте сортировку согласно варианту

### Вариант 1

- линейный выбор;
- сортировка бинарным деревом;
- сортировка Шелла (вариант чисел Фибоначи).

### Вариант 2

- линейный выбор с обменом;
- пирамидальная сортировка;
- поразрядная сортировка.

### **Вариант 3**

- линейный выбор с подсчетом;
- вариант Синглтона быстрой сортировки;
- карманная сортировка.

### **Вариант 4**

- парный обмен;
- быстрая сортировка (выбор медианы с проверкой на упорядоченность);
- квадратичный выбор.

### **Вариант 5**

- стандартный обмен («метод пузырька»);
- поразрядная сортировка;
- сортировка бинарным деревом.

### **Вариант 6**

- просеивание (челночная сортировка);
- сортировка подсчетом;
- пирамидальная сортировка.

### **Вариант 7**

- линейная вставка;
- сортировка слиянием;
- быстрая сортировка (с выбором медианы при помощи RANDOM).

### **Вариант 8**

- сортировка Шелла;
- квадратичный выбор;
- сортировка простыми вставками с использованием бинарного поиска места для нового элемента.

### **Вариант 9**

- линейный выбор;
- карманная сортировка;
- линейный выбор с одновременным упорядочиванием с обеих сторон (поиск минимальных и максимальных элементов одновременно).

### **Вариант 10**

- пузырьковая сортировка;
- сортировка Шелла (вариант чисел Фибоначчи);
- быстрая сортировка (с выбором медианы при помощи RANDOM).

**Задание 2.** Реализуйте демонстрационный вариант с выводом на экран промежуточных результатов для каждой сортировки (для 10–20 элементов).

**Задание 3.** Произведите все свои три сортировки для следующих объемов данных: 100, 1 000, 10 000, 50 000 элементов (для одного и того же процессора) для следующих видов исходных данных:

- а) Random;
- б) упорядоченный массив;
- в) обратно упорядоченный;
- г) почти упорядоченный (10 % элементов стоят не на своих местах);
- д) полуупорядоченный (первая половина массива упорядочена, вторая тоже).

Для каждого из вариантов упорядоченности произведите 100 тестов. Результаты запишите в таблицу.

**Задание 4.** Сделайте выводы.

Обобщите свои знания по теме методы сортировки данных:

- виды сортировок;
- временная сложность методов сортировок;
- в каком случае (в зависимости от исходного массива данных) выгодна та или иная сортировка (исходя из вашего опыта).

## **Лабораторная работа № 6. Способы представления графов**

Во всех задачах данные о графе помещаются в файлах (входном и выходном).

Задание:

1. Представление графа: матрица смежности.  
Граф: ориентированный и неориентированный.  
Задача: Преобразуйте в матрицу инцидентий.
2. Представление графа: матрица смежности.  
Граф: ориентированный и неориентированный.  
Задача: Преобразуйте в список смежности.
3. Представление графа: матрица смежности.  
Граф: ориентированный и неориентированный.  
Задача: Преобразуйте в массив дуг.
4. Представление графа: массив дуг.  
Граф: ориентированный и неориентированный.  
Задача: Преобразуйте в матрицу смежности.

5. Представление графа: массив дуг.  
Граф: ориентированный и неориентированный.  
Задача: Преобразуйте в матрицу инцидентий.
6. Представление графа: массив дуг.  
Граф: ориентированный и неориентированный.  
Задача: Преобразуйте в список смежности.
7. Представление графа: списки смежности.  
Граф: ориентированный и неориентированный.  
Задача: Преобразуйте в матрицу смежности.
8. Представление графа: списки смежности.  
Граф: ориентированный и неориентированный.  
Задача: Преобразуйте в матрицу инцидентий.
9. Представление графа: списки смежности.  
Граф: ориентированный и неориентированный.  
Задача: Преобразуйте в массив дуг.
10. Представление графа: матрица инцидентий.  
Граф: ориентированный и неориентированный.  
Задача: Преобразуйте в матрицу смежности.
11. Представление графа: матрица инцидентий.  
Граф: ориентированный и неориентированный.  
Задача: Преобразуйте в массив дуг.
12. Представление графа: матрица инцидентий.  
Граф: ориентированный и неориентированный.  
Задача: Преобразуйте в список смежности.

**Лабораторная работа № 7. Алгоритмы на графах. Алгоритмы обхода графа. Построение остова минимального веса. Выделение долей в двудольном графе. Точки сочленения графа**

Во всех задачах данные о графе помещаются в файле. Используйте то представление графа, которое указано в задаче. При сдаче задачи укажите, какие «классические алгоритмы теории графов» использовались при решении.

1. Представление графа: матрица смежности.  
Задача: Вы участвуете в конструировании космической станции как специалист в теории графов. Перед Вами поставлена задача: обеспечить дублирование в системе коммуникаций станции для того, чтобы обеспечить

бесперебойную работу в случае выхода из строя одного из узлов системы коммуникаций. Использовать при этом минимальное число дополнительных переходов между узлами системы.

## 2. Представление графа: матрица инцидентий.

Задача: В лесу имеется  $N$  муравейников красных муравьев. Некоторые муравейники соединены друг с другом муравьиными тропами. Муравьи-разведчики сообщили о возможном вторжении на территорию красных муравьев их извечных соперников – черных муравьев. Определите, какой из муравейников должен стать командным центром, чтобы за минимальное время можно было добраться по тропам до остальных муравейников.

## 3. Представление графа: списки смежности.

Задача: Поступила достоверная информация о том, что на острове, где Вы находитесь, произойдет извержение вулкана, в результате чего остров полностью исчезнет. Вы имеете в распоряжении 1 легкий самолет с грузоподъемностью  $H$ . Естественное желание вывезти с острова наиболее ценные предметы: всего имеется  $N$  предметов с весами  $W_i$  и оценочной стоимостью  $C_i$ . Оцените, какие из предметов Вам необходимо взять с собой, чтобы их суммарная стоимость была максимальна? (естественно, что вес не должен превышать  $H$  – грузоподъемность самолета). Число предметов не выше 100 и не менее 1. Примечание: вес, стоимость и грузоподъемность – целочисленные величины из диапазона от 1 до 1 000 кг).

## 4. Представление графа: массив дуг.

Задача: В городе прямоугольной формы, расположенном в холмистой местности, все улицы идут либо с юга на север ( $N$  улиц), либо с востока на запад ( $M$  улиц) так, что все кварталы являются квадратами со стороной 1. Каждый из участков улиц между соседними перекрестками имеет либо только спуск, либо только подъем, либо горизонтален. Матрица  $H[y, x]$  (размера  $M \times N$ ) определяет высоту перекрестков над уровнем моря.

ЗАДАНИЕ : Напишите программу, которая:

- 1) считывает из файла размерность матрицы (числа  $M$  и  $N$ ) и вводит элементы матрицы  $H[i, j]$ ,  $i = \overline{1, M}$ ,  $j = \overline{1, N}$ ;
- 2) запрашивает координаты  $A$  и  $B$ ;
- 3) выдает ответ на вопрос, можно ли спуститься из  $A$  в  $B$  или из  $B$  в  $A$ , непрерывно двигаясь под уклон. При положительном ответе на вопрос:
- 4) определяет хотя бы один такой маршрут движения (и выдает на экран координаты его перекрестков);
- 5) Определяет все такие маршруты.

**5. Представление графа: матрица инциденций.**

Задача: Дана *прямоугольная решетка* с закрашенными и незакрашенными клетками, требуется найти незакрашенный прямоугольник наибольшей площади, вписанный в решетку. Размеры решетки не более чем  $10 \times 10$ .

**6. Представление графа: списки смежности.**

Задача: Найдите *кратчайший путь* передвижения *ладьи* по заданному прямоугольному клеточному полю, соединяющий две указанные его клетки. На поле могут быть препятствия.

**7. Представление графа: массив дуг.**

Задача: *лабиринт* может быть задан матрицей соединений комнат  $A$ , такой, что  $A[i, j] = 1$ , если комната  $i$  соединена с комнатой  $j$  коридором, и  $A[i, j] = 0$  в противном случае. Дана матрица соединений для лабиринта, состоящего из  $n$  комнат, и заданы номера комнат  $i$  и  $j$  ( $\geq 1$  и  $\leq n$ ). Постройте кратчайший путь из комнаты с номером  $i$  в комнату с номером  $j$ .

**8. Представление графа: матрица инциденций.**

Задача: Задана *система односторонних дорог* между  $n$  городами. Определите можно ли проехать из города  $A$  в город  $B$  таким образом, чтобы посетить город  $C$  и не проезжать никакой дороги более одного раза.

**9. Представление графа: списки смежности.**

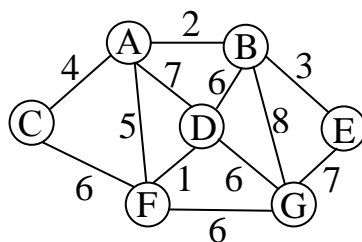
Задача: В *системе двусторонних дорог* за проезд каждой дороги взимается пошлина. Найдите путь из города  $A$  в город  $B$  с минимальной суммой  $S + P$ , где  $S$  – сумма длин дорог пути, а  $P$  – сумма пошлин проезжаемых дорог.

**10. Представление графа: массив дуг.**

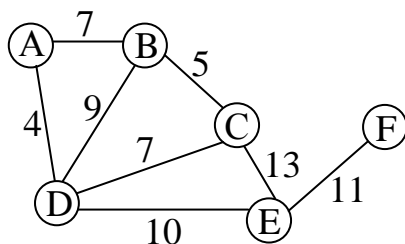
Задача: Имеется  $n$  районов в городе. Найдите, в каком районе необходимо расположить *пункт скорой помощи*, чтобы поездка в самый удаленный по времени район была минимальной. Задано время пути из района  $i$  в район  $j$ , причем время на обратный путь может отличаться от времени прямого пути. Естественно предположить, что поездка по вызову в район  $j$  из района  $i$  займет время на поездку туда и обратно, т.е.  $A[i, j] + A[j, i]$ .

**Лабораторная работа № 8. Построение  
минимального покрывающего дерева (МПД)**

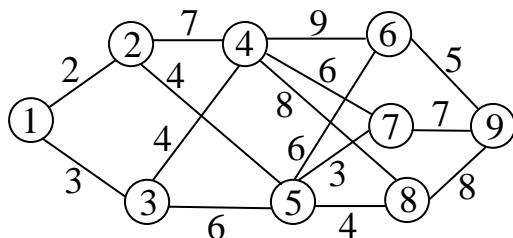
**Вариант 1.** Телевизионная компания планирует подключение к своей кабельной сети несколько новых районов. На рисунке показана структура планируемой сети и расстояние между районами и телецентром, находящемся в точке  $A$ . Необходимо спланировать наиболее экономическую кабельную сеть (вывести минимальную сумму расстояний).



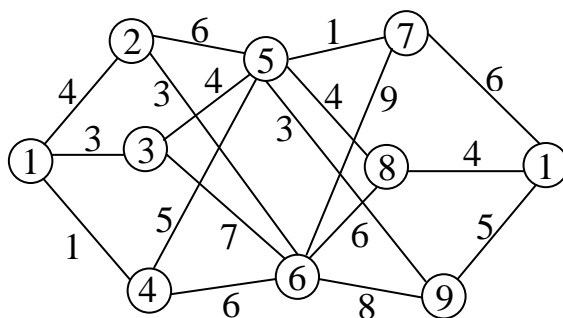
**Вариант 2.** Имеется 6 городов, которые нужно объединить в единую телефонную сеть. Для этого достаточно проложить 5 телефонных линий между городами. Как соединить города так, чтобы суммарная стоимость соединений (телефонного кабеля) была минимальна (и вывести эту стоимость)?



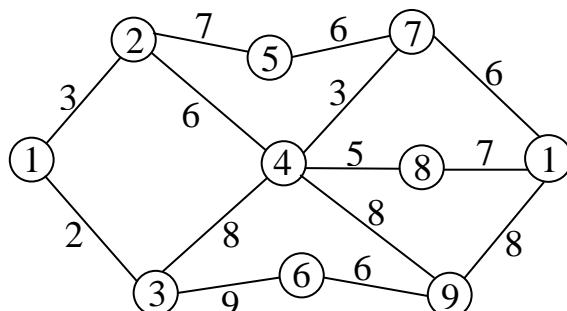
**Вариант 3.** Молочный комбинат осуществляет поставку своей продукции в ряд торговых точек города автомобильным транспортом. Требуется определить маршрут минимальной длины доставки продукции во все торговые точки. Расстояния между торговыми точками известны и представлены на рисунке.



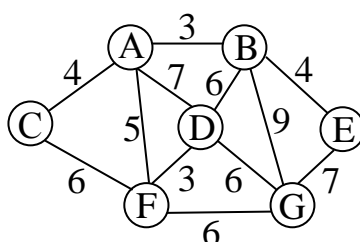
**Вариант 4.** Телевизионная компания планирует подключение к своей кабельной сети нескольких новых районов. На рисунке показана структура планируемой сети и расстояние между районами и телецентром, находящемся в точке 1. Необходимо спланировать наиболее экономичную кабельную сеть (вывести минимальную сумму расстояний).



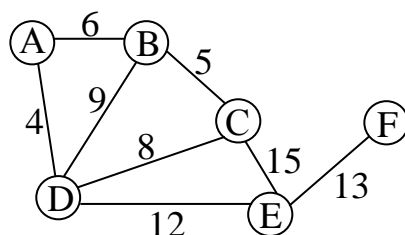
**Вариант 5.** Имеется 10 городов, которые нужно объединить в единую телефонную сеть. Для этого достаточно проложить 9 телефонных линий между городами. Как соединить города так, чтобы суммарная стоимость соединений (телефонного кабеля) была минимальна (и вывести эту стоимость)?



**Вариант 6.** Телевизионная компания планирует подключение к своей кабельной сети нескольких новых районов. На рисунке показана структура планируемой сети и расстояние между районами и телецентром, находящемся в точке A. Необходимо спланировать наиболее экономическую кабельную сеть (вывести минимальную сумму расстояний).

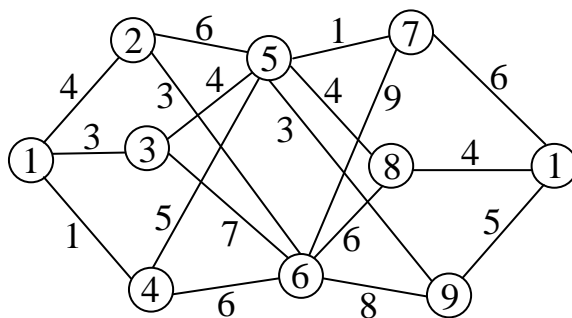


**Вариант 7.** Имеется 6 городов, которые нужно объединить в единую телефонную сеть. Для этого достаточно проложить 5 телефонных линий между городами. Как соединить города так, чтобы суммарная стоимость соединений (телефонного кабеля) была минимальна (и вывести эту стоимость)?

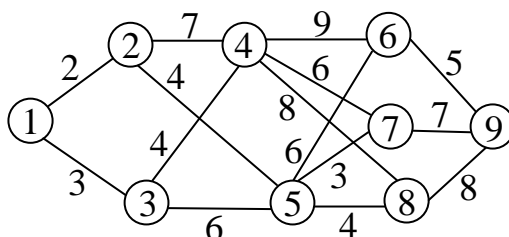


**Вариант 8.** Молочный комбинат осуществляет поставку своей продукции в ряд торговых точек города автомобильным транспортом. Требуется определить маршрут минимальной длины доставки продукции во все торговые точки. Расстояния между торговыми точками известны и представлены на рисунке.





**Вариант 9.** Телевизионная компания планирует подключение к своей кабельной сети нескольких новых районов. На рисунке показана структура планируемой сети и расстояние между районами и телецентром, находящемся в точке 1. Необходимо спланировать наиболее экономическую кабельную сеть (вывести минимальную сумму расстояний).



**Вариант 10.** Имеется 7 городов, которые нужно объединить в единую телефонную сеть. Для этого достаточно проложить 6 телефонных линий между городами. Как соединить города так, чтобы суммарная стоимость соединений (телефонного кабеля) была минимальна (и вывести эту стоимость)?

