

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧЕРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.Г.ШУХОВА»
(БГТУ им.В.Г.Шухова)**

Кафедра программного обеспечения вычислительной техники и
автоматизированных систем

Лабораторная работа №2.1
дисциплина: Дискретная математика
тема: «Комбинаторные объекты»

Выполнил: ст. группы ПВ-201
Машуров Дмитрий Русланович
Проверил: Бондаренко Т.В.

Белгород 2021

Лабораторная работа № 2.1

Алгоритмы порождения комбинаторных объектов

Цель работы: изучить основные комбинаторные объекты, алгоритмы их порождения, программно реализовать и оценить временную сложность алгоритмов.

Задания

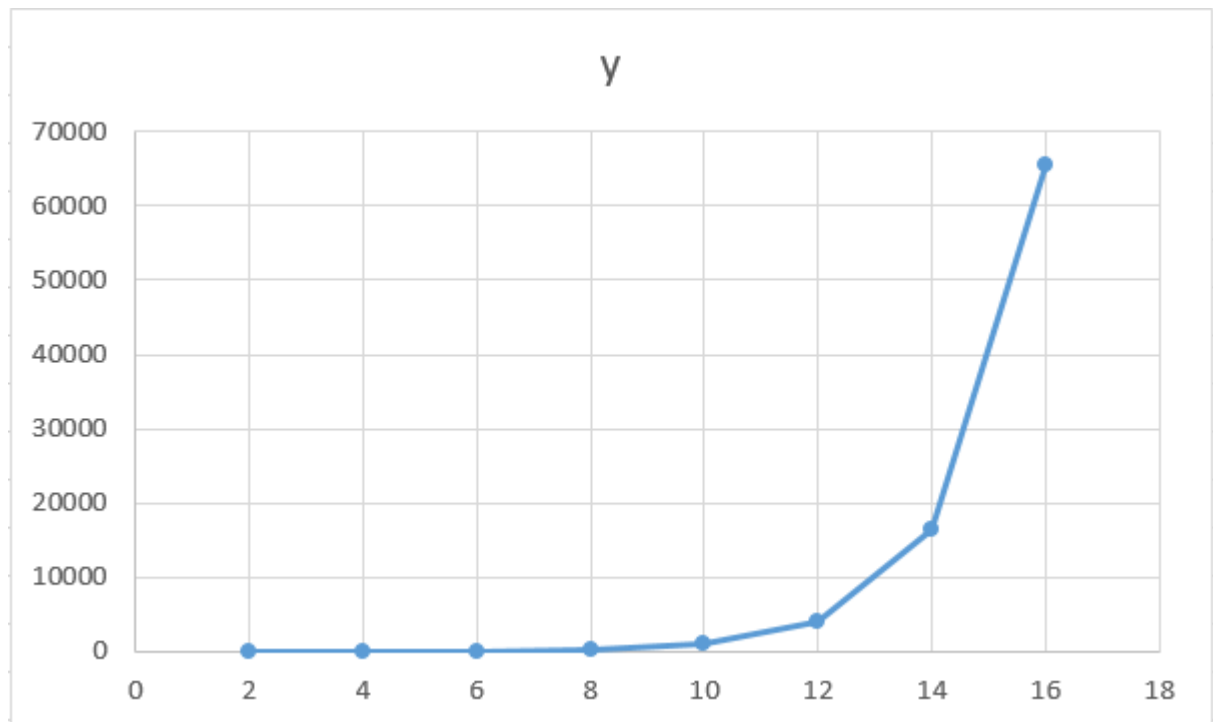
1. Реализовать алгоритм порождения подмножеств.
2. Построить график зависимости количества всех подмножеств от мощности множества.
3. Построить графики зависимости времени выполнения алгоритмов п.1 на вашей ЭВМ от мощности множества.
4. Определить максимальную мощность множества, для которого можно получить все подмножества не более чем за час, сутки, месяц, год на вашей ЭВМ.
5. Определить максимальную мощность множества, для которого можно получить все подмножества не более чем за час, сутки, месяц, год на ЭВМ, в 10 и в 100 раз быстрее вашей.
6. Реализовать алгоритм порождения сочетаний.
7. Построить графики зависимости количества всех сочетаний из n по k от k при $n = (5, 7, 9)$.
8. Реализовать алгоритм порождения перестановок.
9. Построить график зависимости количества всех перестановок от мощности множества.
10. Построить графики зависимости времени выполнения алгоритма п.8 на вашей ЭВМ от мощности множества.
11. Определить максимальную мощность множества, для которого можно получить все перестановки не более чем за час, сутки, месяц, год на вашей ЭВМ.
12. Определить максимальную мощность множества, для которого можно получить все перестановки не более чем за час, сутки, месяц, год на ЭВМ, в 10 и в 100 раз быстрее вашей.
13. Реализовать алгоритм порождения размещений. 14. Построить графики зависимости количества всех размещений из n по k от k при $n = (5, 7, 9)$

Выполнение:

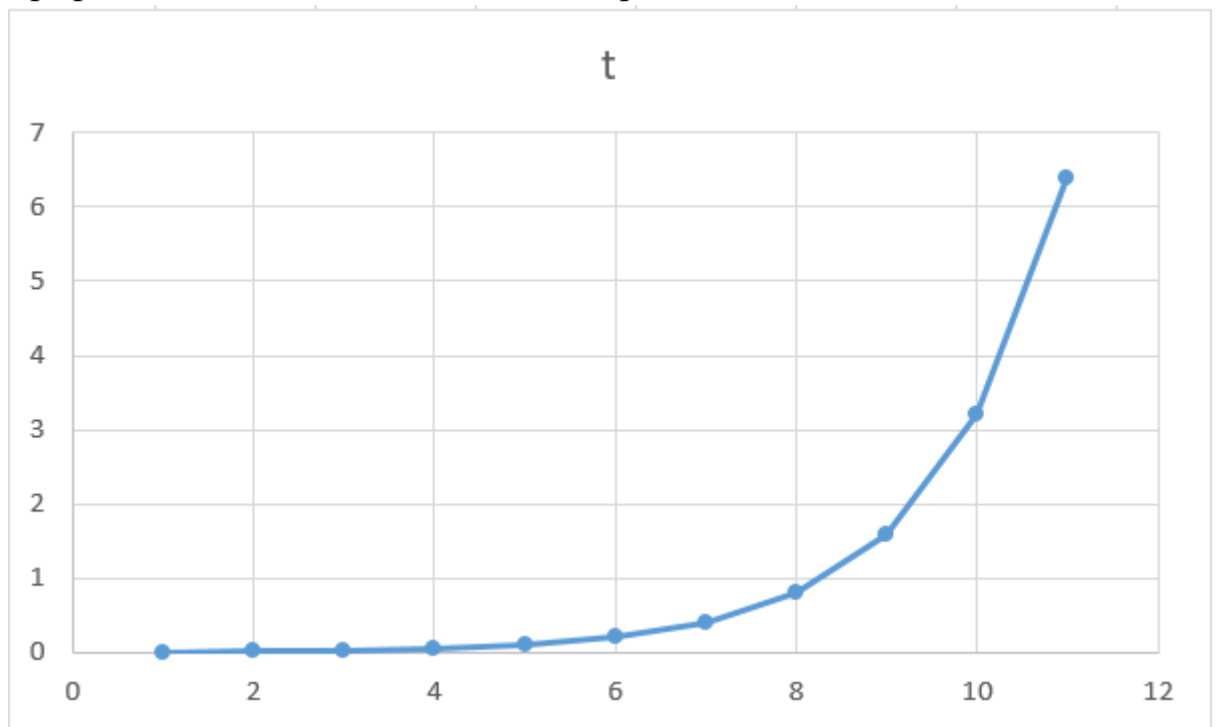
1. Реализация подмножеств:

```
static void get_subsets_inner(int *a, int n, int i) {  
    static int d[100];  
  
    for (int x = 0; x < 2; ++x) {  
        d[i] = x;  
  
        if (i == n - 1) {  
            output_array_with_mask(a, n, d);  
        } else {  
            get_subsets_inner(a, n, i + 1);  
        }  
    }  
}
```

2. График



3. График зависимости выполнения алгоритма на моей ЭВМ



При мощности множества 8 затрачивается $\approx 0,83$ сек, следовательно:

$$t \approx 0,8 * 2^{n-8}$$

Тогда:

$$n \approx \log_2(t * 1,25) + 8$$

4. Определяю максимальную мощность множества, для которого можно получить все подмножества не более чем:

- За час:

$$n \approx \log_2(3600 * 1,25) + 8 \approx 21$$

- За сутки:

$$n \approx \log_2(24 * 3600 * 1,25) + 8 \approx 25$$

- За месяц:

$$n \approx \log_2(30 * 24 * 3600 * 1,25) + 8 \approx 30$$

- За год:

$$n \approx \log_2(12 * 30 * 24 * 3600 * 1,25) + 8 \approx 34$$

5. Определим максимальную мощность множества, для которого можно получить все подмножества не более чем за час, сутки, месяц, год на ЭВМ, в 10 и в 100 раз медленнее

Получу формулу:

$$t \approx \frac{0,8 * 2^{n-8}}{10} - \text{для ЭВМ, которая в 10 раз медленнее}$$

$$t \approx \frac{0,8 * 2^{n-8}}{100} - \text{для ЭВМ, которая в 100 раз медленнее}$$

Соответственно:

$$n \approx \log_2(t * 12,5) + 8 - \text{для первой формулы}$$

$$n \approx \log_2(t * 125) + 8 - \text{для второй формулы}$$

- За час:
 - В 10 раз: $n \approx \log_2(t * 12,5) + 8 \approx 24$
 - В 100 раз: $n \approx \log_2(t * 125) + 8 \approx 27$
- За сутки:
 - В 10 раз: $n \approx \log_2(t * 12,5) + 8 \approx 28$
 - В 100 раз: $n \approx \log_2(t * 125) + 8 \approx 32$
- За месяц:
 - В 10 раз: $n \approx \log_2(t * 12,5) + 8 \approx 33$
 - В 100 раз: $n \approx \log_2(t * 125) + 8 \approx 37$
- За год:
 - В 10 раз: $n \approx \log_2(t * 12,5) + 8 \approx 37$
 - В 100 раз: $n \approx \log_2(t * 125) + 8 \approx 40$

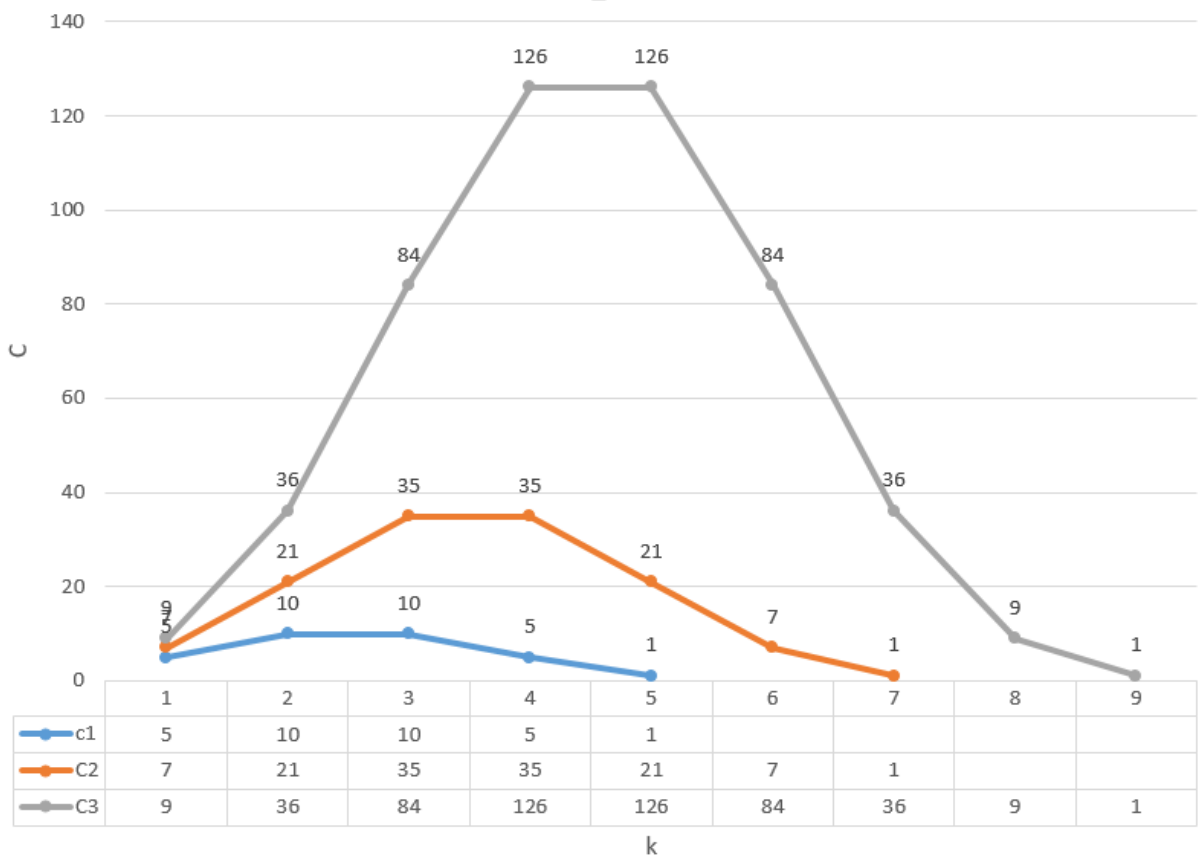
6. Реализовать алгоритм порождения сочетаний

```
void get_combinations_inner(int a[], int n, int i, int k, int b) {
    static int c[100];

    for (int j = b; j < n; ++j) {
        c[i] = a[j];

        if (i == k - 1) {
            output_arr(c, k);
        } else {
            get_combinations_inner(a, n, i + 1, k, j + 1);
        }
    }
}
```

7. Построить графики зависимости количества всех сочетаний из n по k от k при $n = (5, 7, 9)$



8. Реализовать алгоритм порождения перестановок

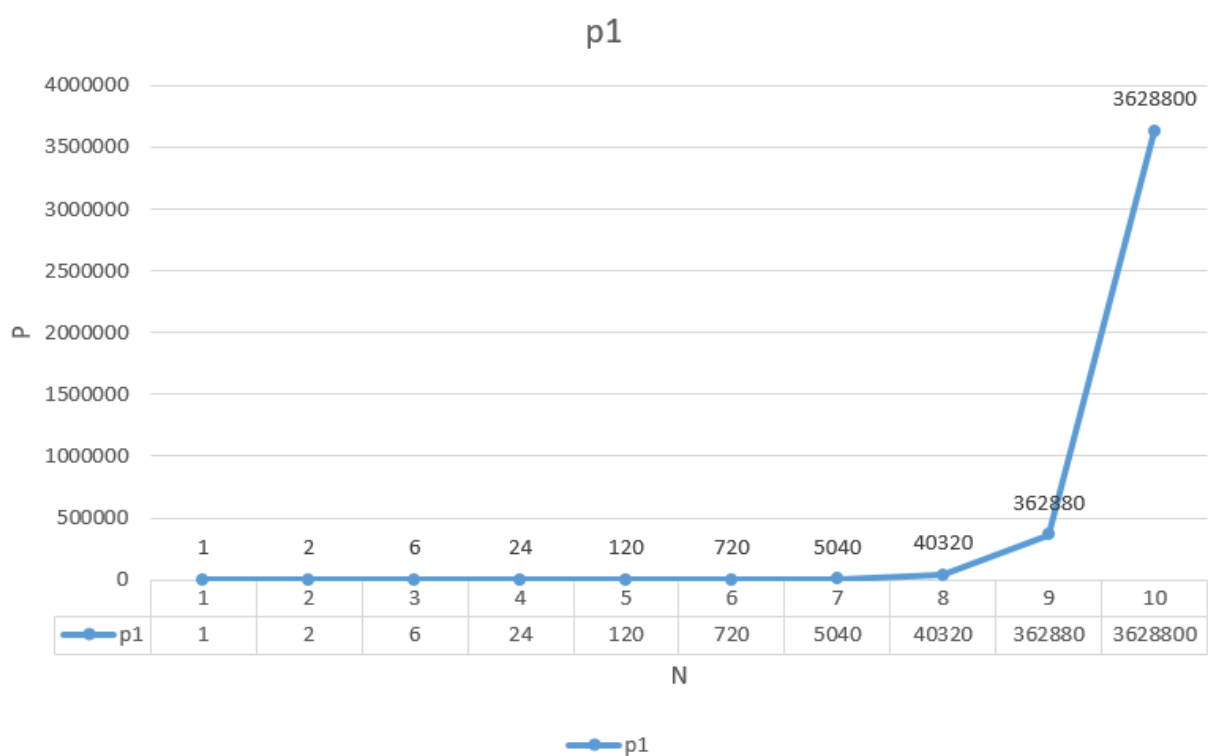
```
static void get_permutations_inner(int a[], int n, int i) {
    static int m[100];
    static int t[100];

    for (int j = 0; j < n; ++j) {
        if (t[j] == 0) {
            m[i] = a[j];
            t[j] = 1;

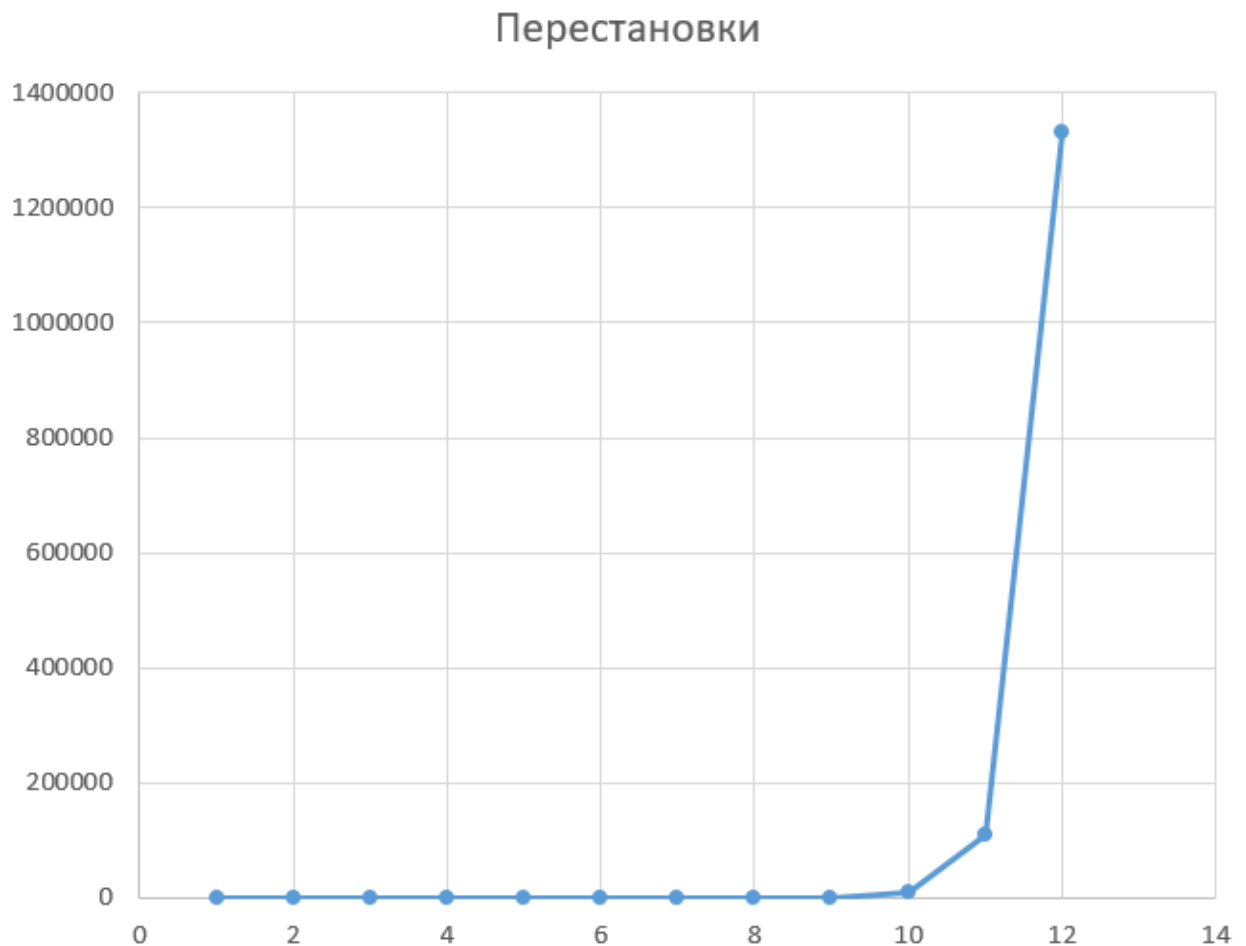
            if (i == n - 1) {
                output_arr(m, n);
            } else {
                get_permutations_inner(a, n, i + 1);
            }

            t[j] = 0;
        }
    }
}
```

9. Построить график зависимости количества всех перестановок от мощности множества



10. Построить графики зависимости времени выполнения алгоритма п.8 на вашей ЭВМ от мощности множества.



Определяю формулу для вычисления времени t :

$$t \approx 2 * \frac{n!}{6!} = 2 * \frac{n!}{720} = \frac{n!}{360}, \text{ где } n - \text{мощность множества}$$

Следовательно:

$$n = (360 * t)!?$$

11. Определить максимальную мощность множества, для которого можно получить все перестановки не более чем за час, сутки, месяц, год на вашей ЭВМ.

- Не более чем за час: 10
- Не более чем за сутки: 11
- Не более чем за месяц: 13
- Не более чем за год: 14

12. Определить максимальную мощность множества, для которого можно получить все перестановки не более чем за час, сутки, месяц, год на ЭВМ, в 10 и в 100 раз быстрее вашей.

Выражу формулу для вычисления:

$$t \approx \frac{n!}{3600} - \text{для ЭВМ, которая в 10 раз мощнее}$$

$$t \approx \frac{n!}{36000} - \text{для ЭВМ, которая в 100 раз мощнее}$$

Соответственно:

$$n = (3600 * t)?! - \text{для ЭВМ, которая в 10 раз мощнее}$$

$$n = (36000 * t)?! - \text{для ЭВМ, которая в 100 раз мощнее}$$

- Не более чем за час
 - В 10 раз мощнее
 - В 100 раз мощнее
- Не более чем за сутки
 - В 10 раз мощнее
 - В 100 раз мощнее
- Не более чем за месяц
 - В 10 раз мощнее
 - В 100 раз мощнее
- Не более чем за год
 - В 10 раз мощнее
 - В 100 раз мощнее

13. Реализовать алгоритм порождения размещений. 14. Построить графики зависимости количества всех размещений из n по k от k при $n = (5, 7, 9)$

Реализация:

```
void get_accommodation_inner(int a[], int n, int k, int i) {
    static int o[100];
    static int t[100];

    for (int j = 0; j < n; ++j) {
        if (t[j] == 0) {
            o[i] = a[j];
            t[j] = 1;

            if (i == k - 1) {
                output_arr(o, k);
            } else {
                get_accommodation_inner(a, n, k, i + 1);
            }

            t[j] = 0;
        }
    }
}
```

График:

