# Упражнения: Комбинаторни алгоритми

Тествайте решението в Judge: <https://judge.softuni.org/Contests/3641/Combinatorial-Algorithms-Exercises>

Използвайте дадения скелет:

## Пермутация без повторения

Даден ви е **набор** от елементи. Намерете всички **пермутации без повторение**.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| A B C | A B C  A C B  B A C  B C A  C B A  C A B |

### Насоки

За да напишем алгоритъм, трябва да направим дъно на рекурсията. Дъното ще проверява дали **индексът** е **по-голям или равен** на **дължината на пермутацията**:

Graphical user interface, text, application

Description automatically generated

Алгоритъмът започва от 0 до дължината на пермутацията. Ако не сме използвали сегашната пермутация, трябва да:

1. Маркираме **сегашният индекс** като използван, за да сме сигурни, че **не се повтаря**
2. Задаваме текущата **пермутация** като тази от входа, за да актуализираме нашата **последователност**
3. Извикваме GeneratePermutations() с **index+1**, за да отидем на следващата възможна пермутация
4. Маркираме сегашния индекс като неизползван, за да може да го използваме отново за следваща пермутация

Graphical user interface, text, application

Description automatically generated

## Пермутация с повторения

Даден ви е **набори от елементи** с възможност за повтаряне. Намерете **всички пермутации**.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| A B B | A B B  B A B  B B A |

## Вариация без повторения

Даден ви е **набор** от елементи. Намерете **вариациите** от **k елемента без повторение**.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| A B C  2 | A B  A C  B A  B C  C A  C B |

### Насоки

Каква е разликата от предишния алгоритъм? Къде влиза **K**? **Преди** или **по време** на алгоритъма?

## Вариация с повторения

Даден ви е набор от елементи. Намерете всички **вариации** от **k елемента с повторения**.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| A B C  2 | A A  A B  A C  B A  B B  B C  C A  C B  C C |

## Комбинации без повторение

Даден ви е **набор** от елементи. Генерирайте всички **комбинации** от **k елемента** **без повторения**.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| A B C  2 | A B  A C  B C |

### Решение

Алгоритъмът се различава от пермутацията по следния начин:

For-цикъла започва от startIndex. За всяко обхождане задаваме текущата комбинация да е равна на тази от входа и извикваме generate() с **index + 1** и **currentIndex + 1**

Graphical user interface, text, application

Description automatically generated

## Комбинации без повторение

Даден ви е **набор** от елементи. Генерирайте всички **комбинации** от **k елементи с повторение**.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| A B C  2 | A A  A B  A C  B B  B C  C C |

## N избора K пъти

Дадени са ви числата **n** и **k**. Пресметнете всички възможните **комбинации n-k** (**без повторение**).

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 3  2 | 3 |
| 5  3 | 10 |