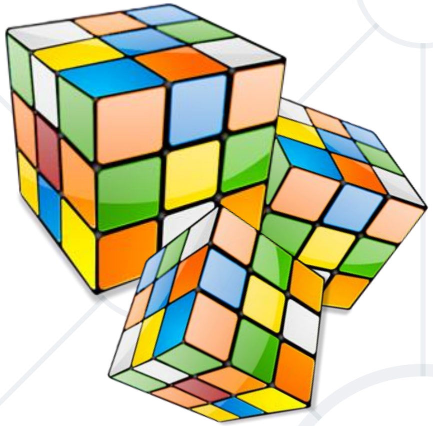
Combinatorial Problems

Permutations, Variations, Combinations and N choose K



SoftUni Team

Technical Trainers

Software University

Software University

https://softuni.bg

Table of Contents

1. Permutations
2. Variations
3. Combinations
4. N Choose K Count

Permutations

Permutations

- Permutation of a set is an arrangement of its members into a sequence or linear order
  - If the set is already ordered, a rearrangement of its elements
- There are two types of permutations
  - Without repetition
  - With repetition

Permutations

- Order A, B and C in all possible ways

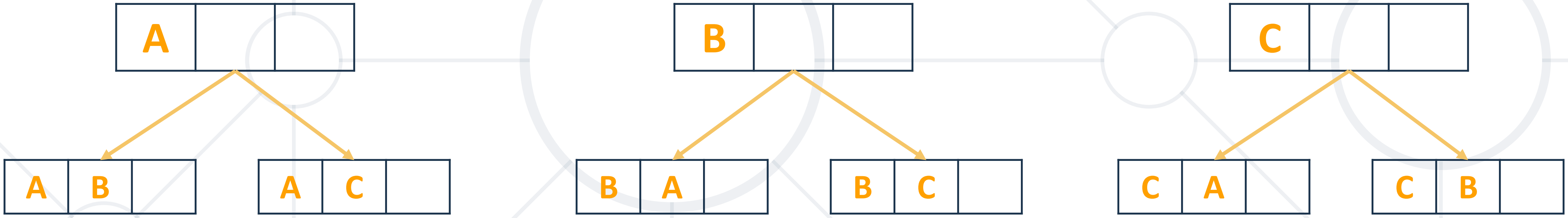
A		

B		

C		

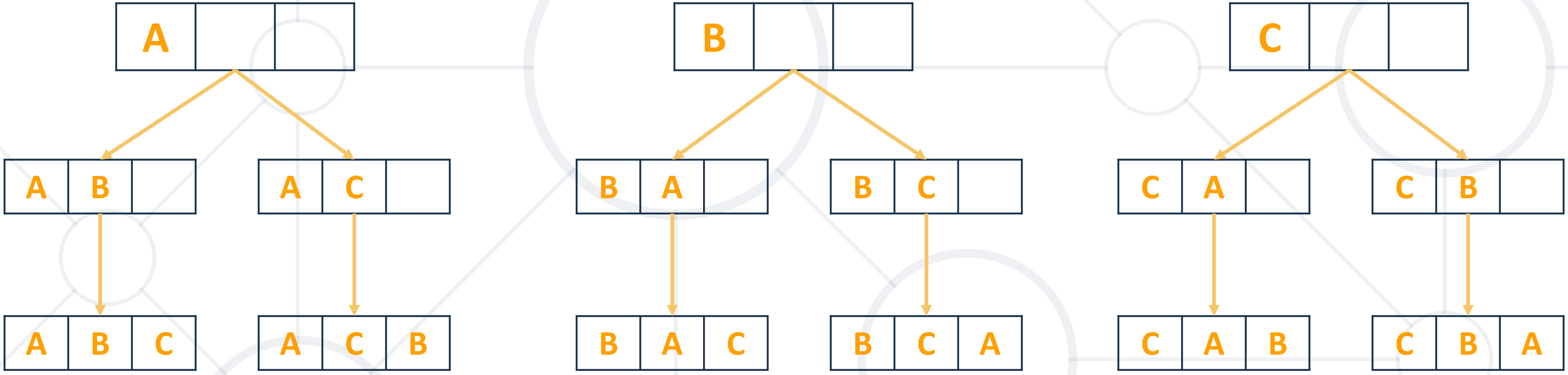
Permutations

- Order **A**, **B** and **C** in all possible ways



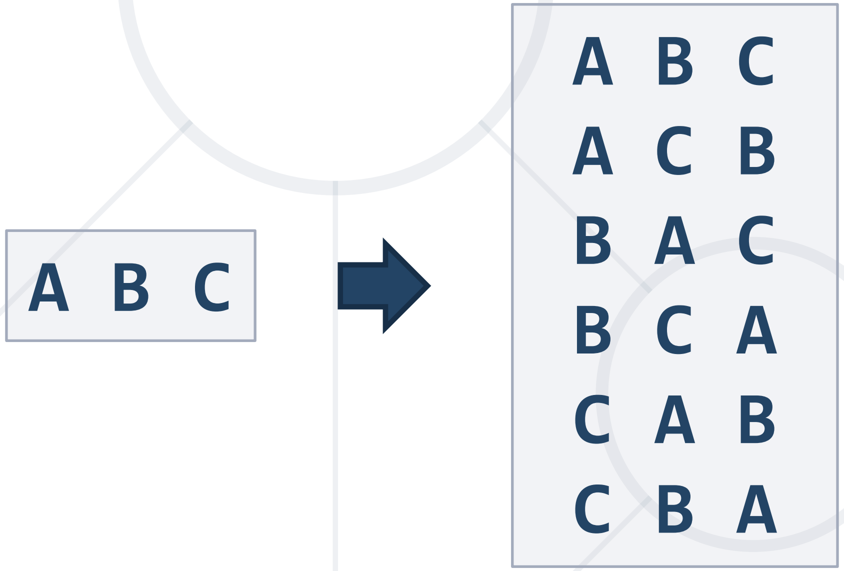
Permutations

- Order **A**, **B** and **C** in all possible ways



Problem: Generate Permutations

- Generates all possible **permutations** of a given set of elements
- You can **pick** each **item only once**

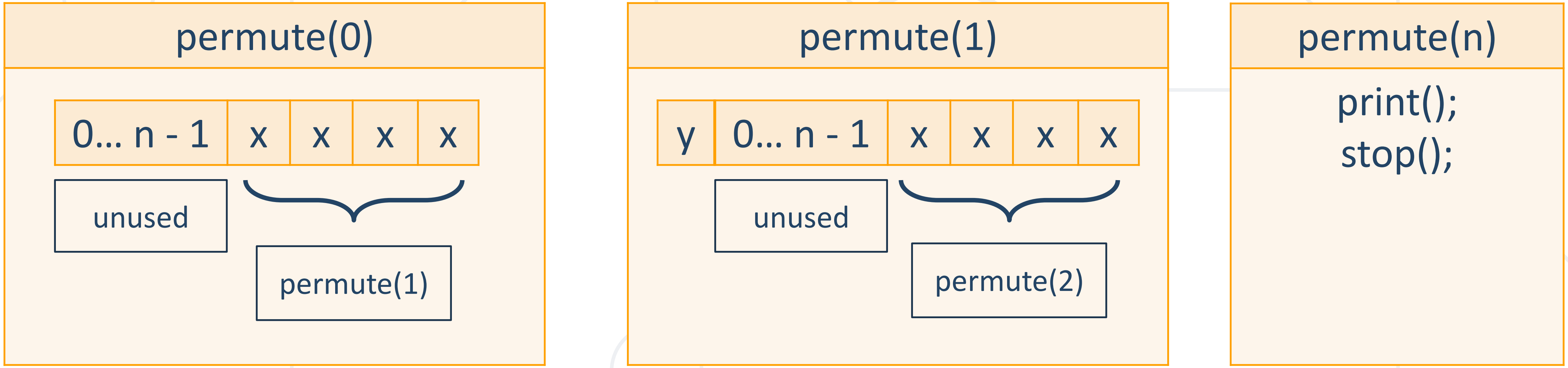


A B C

A B C
A C B
B A C
B C A
C A B
C B A

Algorithm: Permutations

- Algorithm **permute(index)** to generate variations **P(n)**
  - Put **unused** elements **0 … n-1** at position **index**
  - Mark/unmark elements as **being used**
  - Call **permute(index + 1)** to generate the rest of the array



permute(0)

0… n - 1	x	x	x	x

unused

permute(1)

permute(1)

y	0… n - 1	x	x	x	x

unused

permute(2)

permute(n)

print();
stop();

Generating Permutations

```
static void Permute(int index)
{
  if (index >= permutations.Length)
    Print();
  else
    for (int i = 0; i < elements.Length; i++) {
      if (!used[i]) {
        used[i] = true;
        permutations[index] = elements[i];
        Permute(index + 1);
        used[i] = false;
      }
    }
}
```

Permutations Count

- Order **A**, **B** and **C** in all possible ways
- How many ways are there?

A	B	C

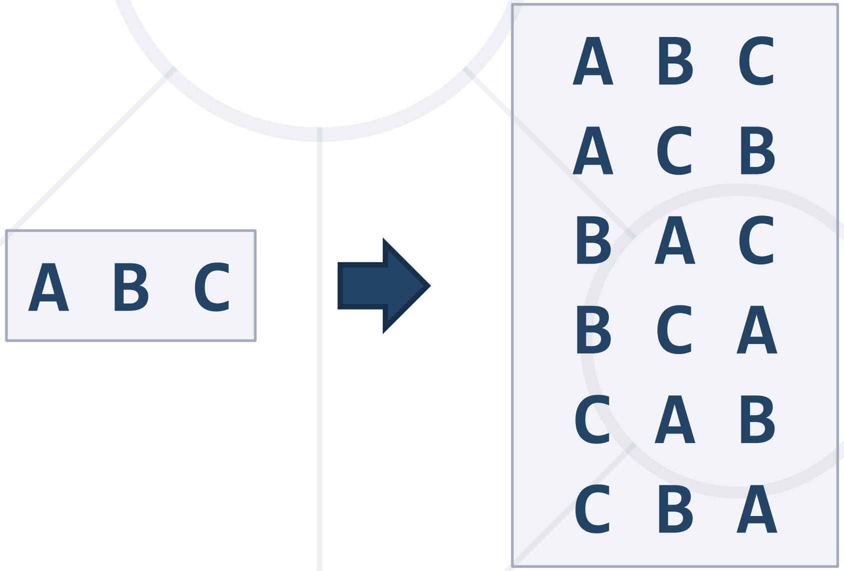
3	2	1

$n! = 3!$

6 possible ways

Problem: Optimize Permutations

- Generates all possible **permutations** of a given set of elements
  - **Without** using **extra memory**



A B C

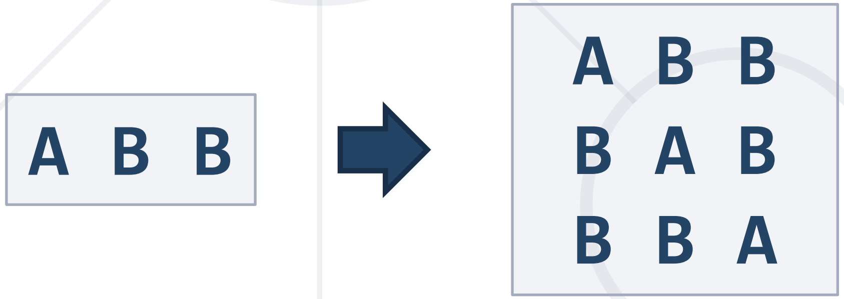
A B C
A C B
B A C
B C A
C A B
C B A

Generating Permutations

```
static void Permute(int index)
{
  if (index >= elements.Length)
    Print();
  else {
    Permute(index + 1);
    for (int i = index + 1; i < elements.Length; i++) {
      Swap(index, i);
      Permute(index + 1);
      Swap(index, i);
    }
  }
}
```

Problem: Permutations with Repetition

- What about `array = new [] { A, B, B }`
- By definition: permutations { A, B', B'' } == { A, B'', B' }
- Generate all permutations from a `multi-set`



A B B

A B B
B A B
B B A

Solution: Permutations with Repetition

```
static void Permute(int index) {
  if (index >= elements.Length)
    Print();
  else {
    Permute(index + 1);
    var swapped = new HashSet<string> { elements[index] };
    for (int i = index + 1; i < elements.Length; i++) {
      if (!swapped.Contains(elements[i])) {
        Swap(index, i);
        Permute(index + 1);
        Swap(index, i);
        swapped.Add(elements[i]);
      }
    }
  }
}
```

Permutations with Repetition Count

- Order **A**, **B** and **B** in all possible ways
- In how many ways we can do that?

A	B	B

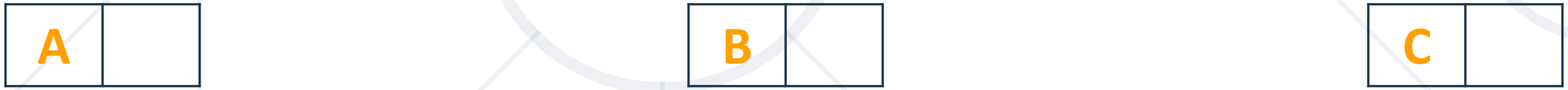
$$\frac{n!}{s1!s2!..sk!} = \frac{3!}{2!1!}$$

3 different ways

Variations

Variations

- Order **A**, **B** and **C** in all possible ways int **k slots**
- **Pick** each **item** only **once**



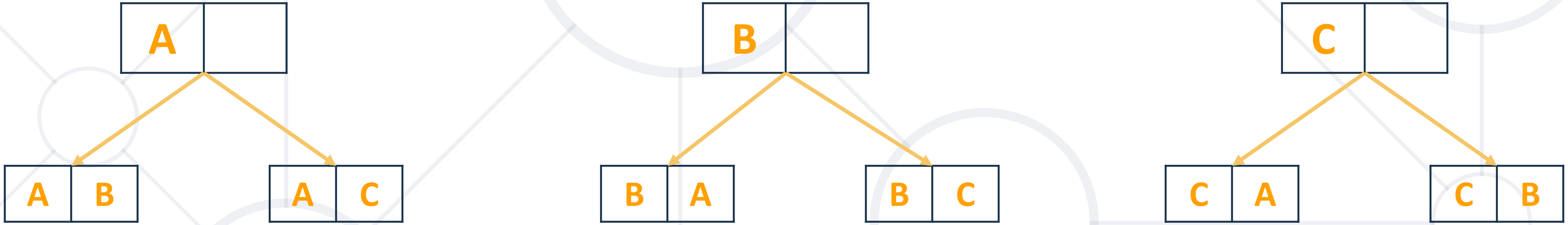
A

B

C

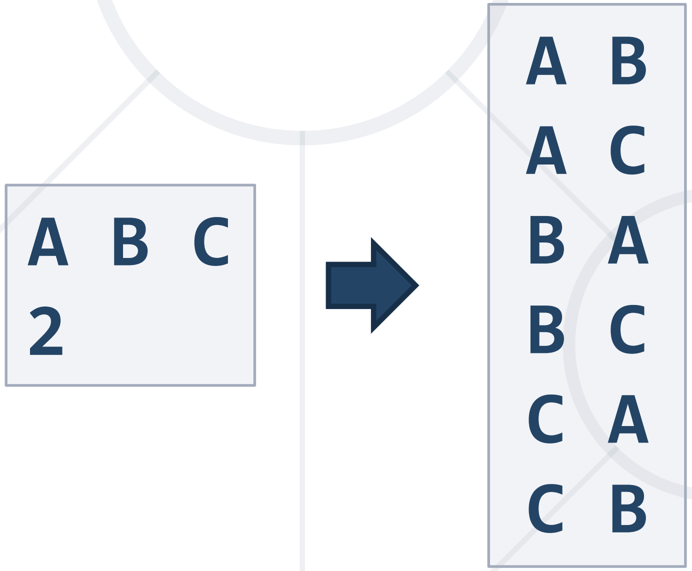
Variations

- Order **A**, **B** and **C** in all possible ways int **k slots**
- **Pick** each **item** only **once**



Problem: Generate Variations

- Generates all possible **variations of k** from a set of elements
- You can **pick** an **item once**



A B C
2

A B
A C
B A
B C
C A
C B

Generating Variations

```
static void Variations(int index)
{
  if (index >= variations.Length)
    Print();
  else
    for (int i = 0; i < elements.Length; i++) {
      if (!used[i]) {
        used[i] = true;
        variations[index] = elements[i];
        Variations(index + 1);
        used[i] = false;
      }
    }
}
```

Variations Count

- **Order two** from **A**, **B**, **C** and **D** in all possible ways
- How many ways are there?

A	B	C	D

4	3

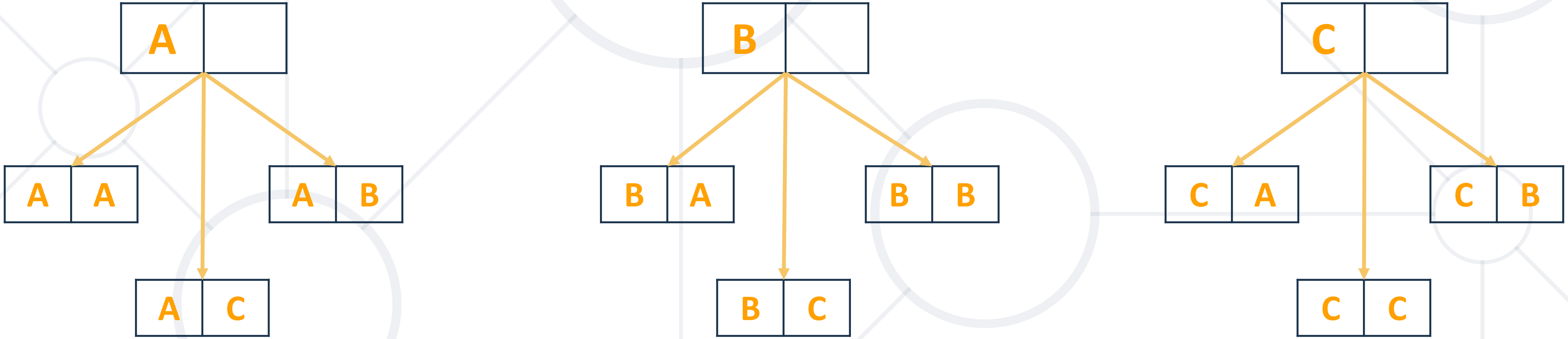
Multiply

$$\frac{n!}{(n-k)!} = \frac{4!}{2!}$$

Twelve different ways

Variations with Repetitions

- Order **A**, **B** and **C** in all possible ways into k slots
- You can **pick** an item **multiple times**



Problem: Generate Variations with Reps

- Generates all possible **variations** of a given elements
  - You can **pick** an **item multiple times**

```
A B C
2
```

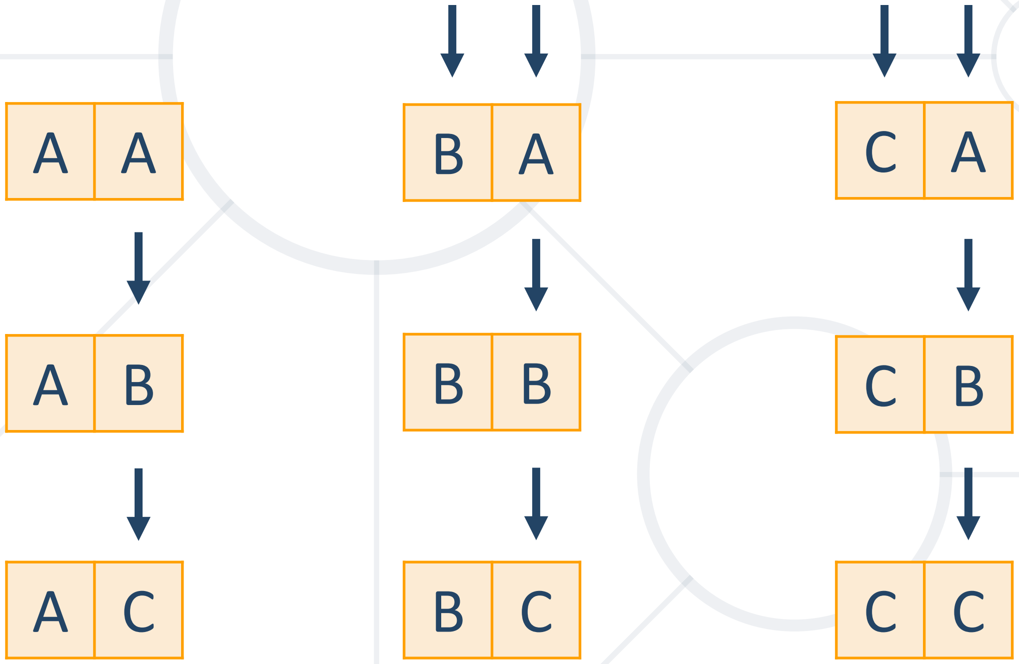
```
A A
A B
A C
B A
B B
B C
C A
C B
C C
```

Generating Permutations

```
static void Variations(int index)
{
  if (index >= variations.Length) {
    Print();
  }
  else {
    for (int i = 0; i < elements.Length; i++) {
      variations[index] = elements[i];
      Variations(index + 1);
    }
  }
}
```

Variations with Reps: Iterative Algorithm

- Generating the variations for ***n*** = ***3*** and ***k*** = ***2***



A A

A B

A C

B A

B B

B C

C A

C B

C C

Variations with Reps: Iterative Algorithm

```
while (true) {
  Print(arr);
  int index = k - 1;
  while (index >= 0 && arr[index] == n-1)
    index--;
  if (index < 0)
    break;
  arr[index]++;
  for (int i = index + 1; i < k; i++)
    arr[i] = 0;
}
```

```
int n = 5;
int k = 3;
int[] arr = new int[k];
```

```
(0, 0, 0)
(0, 0, 1)
    …
(4, 4, 2)
(4, 4, 3)
(4, 4, 4)
```

Variations Count

- **Order two** from **A**, **B**, **C** and **D** in all possible ways
- How many ways are there?

A	B	C	D

4	4

Multiply

$$n^{\mathrm{k}} = 4^2$$

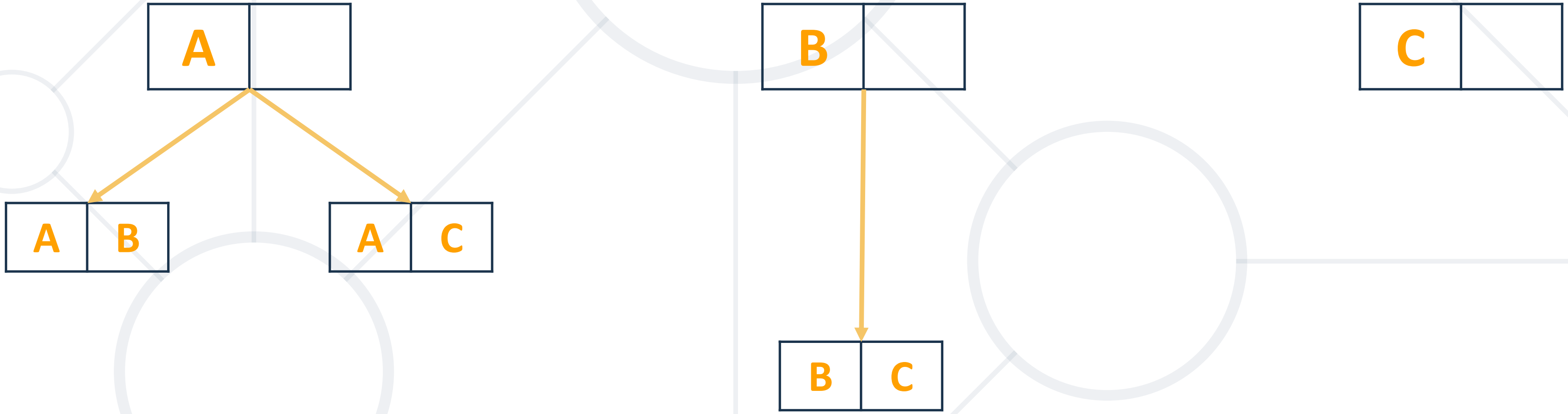
Sixteen different ways

$$\frac{n!}{k!\,(n-k)!}$$

Combinations

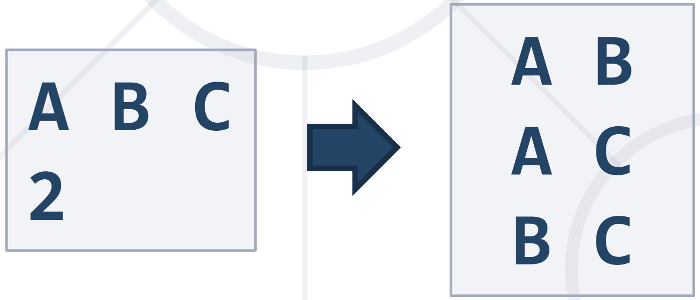
Combinations

- **Pick two** form **A**, **B** and **C**
- Order does not matter



Problem: Generate Combinations

- Generates all possible combinations from a given elements
  - You can **pick** each **item** only **once**



A B C
2

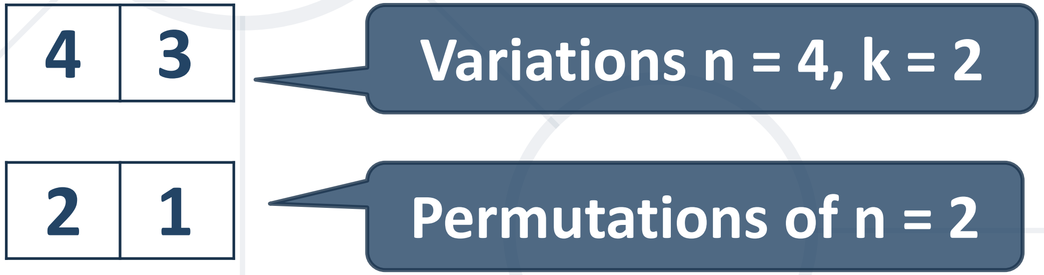
A B
A C
B C

Combinations without Repetition

```
static void Combinations(int index, int start)
{
  if (index >= slots.Length)
    Print();
  else {
    for (int i = start; i < elements.Length; i++) {
      slots[index] = elements[i];
      Combinations(index + 1, i + 1);
    }
  }
}
```

Combinations Count

- Pick two from {**A**, **B**, **C**, **D**} in all possible ways, **order does not matter**
- How many ways are there?



4	3
2	1

Variations n = 4, k = 2

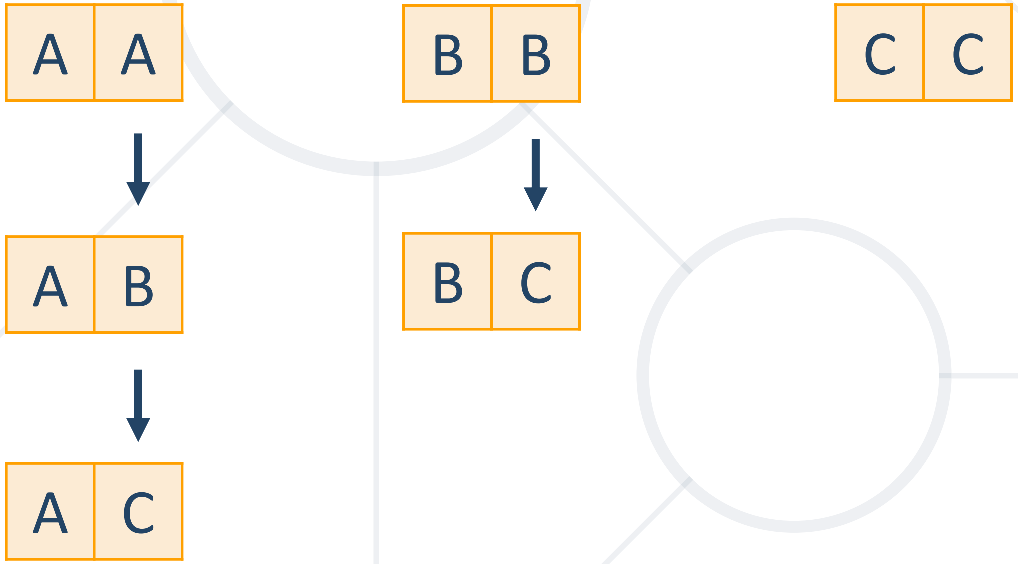
Permutations of n = 2

$$\frac{n!}{k!(n-k)!} = \frac{4!}{2!2!}$$

6 different ways

Algorithm: Combinations with Repetition

- Generating the combinations for ***n*** = ***3*** and ***k*** = ***2***



A A

A B

A C

B B

B C

C C

Generate Combinations with Repetition

```
static void Combinations(int index, int start)
{
  if (index >= slots.Length)
    Print();
  else {
    for (int i = start; i < elements.Length; i++) {
      slots[index] = elements[i];
      Combinations(index + 1, i);
    }
  }
}
```

$$C_n^k = \binom{n}{k} = \frac{n!}{(n-k)!\,k!}$$

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$$

N Choose K Count

Problem: Combinations Count

- How many **combinations** we have when **n = 16**, **k = 15**?
- **Solution**:

$$C_n^k = \binom{n}{k} = \frac{n!}{(n-k)!\,k!}$$

  - How many ways to pick 15 items?

    16 * **15** * **14** * … * **2**

  - Divide by the number of ways in which you can arrange 15 numbers

    15 * 14 * 13 * … * 1

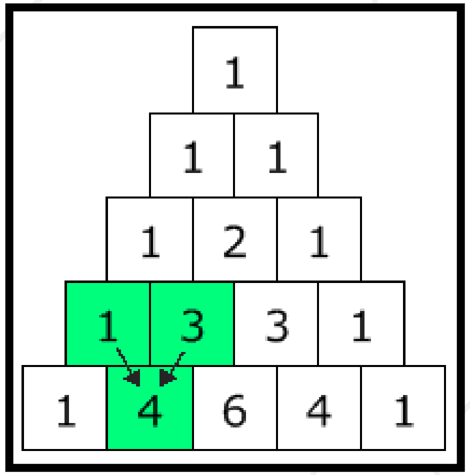
- Possible combinations → **16**

Pascal's Triangle

- In how many ways each **node** can be reached?
- Quickly find **N choose K** count
  - Go **down** to row **n** (the top row is 0)
  - Move along **k** places to the **right**

n = 4

k = 1



1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

Binomial Coefficients: Calculation

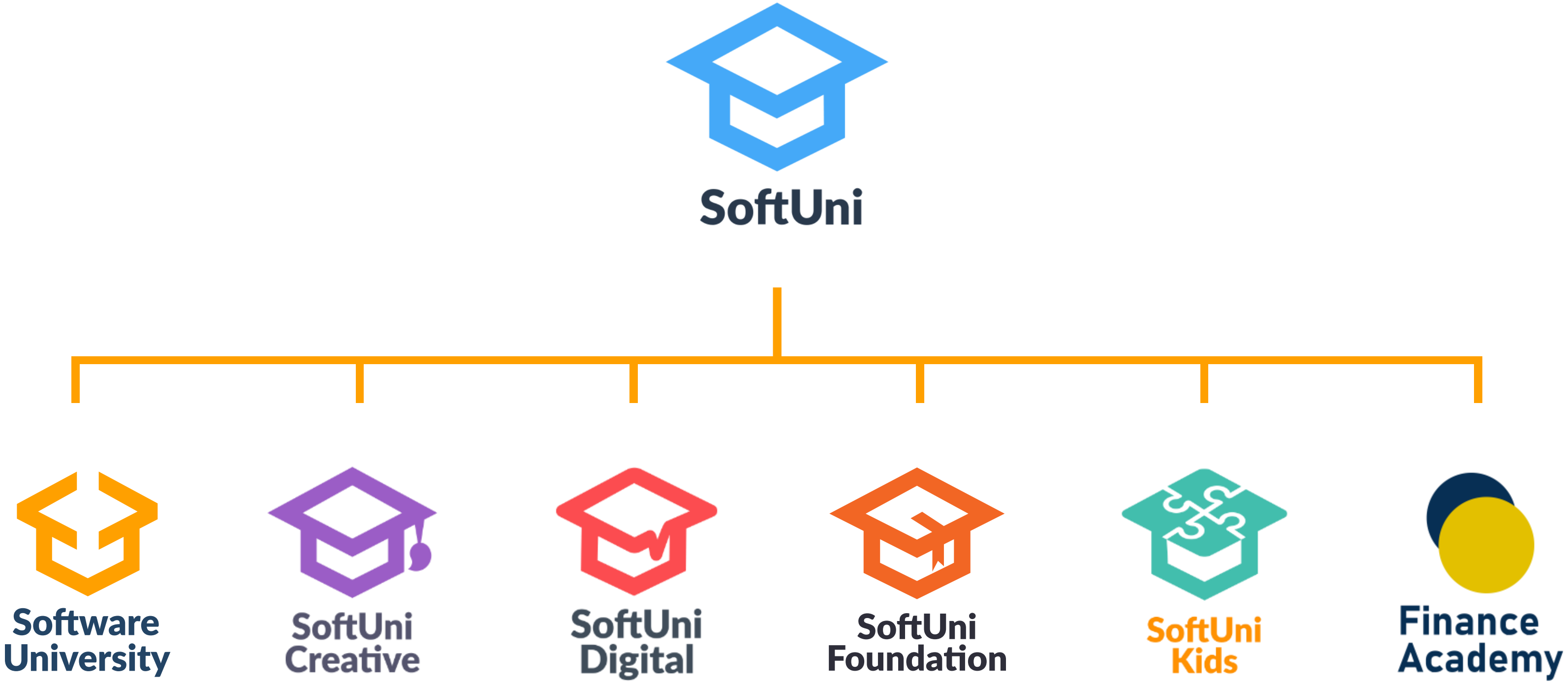
```
static long Binom(int n, int k)
{
  if (n <= 1)
    return 1;
  if (k == 0 || k == n)
    return 1;
  return Binom(n - 1, k) + Binom(n - 1, k - 1);
}
```

Summary

Software University

- **Permutations** – Ways to order n elements
- **Variations** – Ways to **order k** of **n** elements
- **Combinations** – Ways to **choose k** of **n** elements
- Pascal's Triangle
    - Binomial Coefficients – **N choose K Count**

Questions?



SoftUni

Software University

SoftUni Creative

SoftUni Digital

SoftUni Foundation

SoftUni Kids

Finance Academy

SoftUni Diamond Partners

Software University

SUPER HOSTING .BG

Coca-Cola HBC Bulgaria

POKERSTARS
POKER | CASINO | SPORTS
a Flutter International brand

INDEAVR
Serving the high achievers

AMBITIONED

DRAFT KINGS

SOFTWARE GROUP

createX

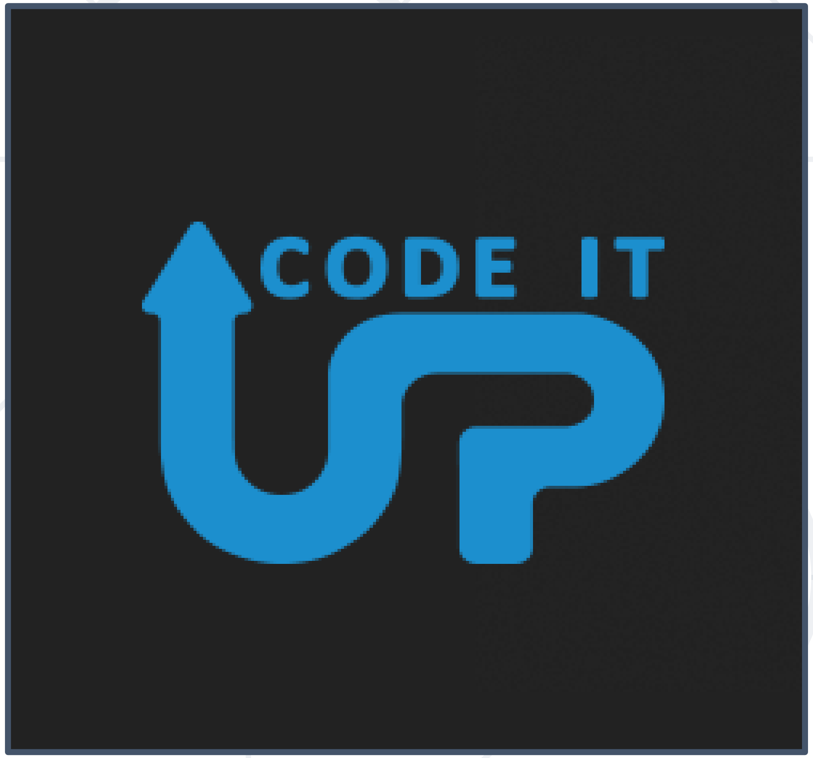
Postbank
Решения за твоето утре

BOSCH

DXC TECHNOLOGY

SmartIT

Educational Partners



License

Trainings @ Software University (SoftUni)

- Software University – High-Quality Education, Profession and Job for Software Developers
  - softuni.bg
- Software University Foundation
  - http://softuni.foundation
- Software University @ Facebook
  - facebook.com/SoftwareUniversity
- Software University Forums
  - forum.softuni.bg