



Міністерство освіти і науки України  
Національний технічний університет України  
“Київський політехнічний інститут імені Ігоря Сікорського”  
Факультет інформатики та обчислювальної техніки  
Кафедра Інформаційних систем і технологій

Лабораторна робота №1  
Теорія алгоритмів  
*«Вступ до алгоритмізації»*  
Варіант 3

Виконали студенти групи ІА-11:	Перевірів:
Бригада “IT CROWD” #3	ас. Цимбал С. І.
Момот А.Р.	
Юрченко В.І	
Старовойтов В.П.	
Щур С. В.	Захищено балом: <i>n</i>

*Мета роботи:* навчитись створювати, описувати та реалізовувати алгоритми.

*Хід роботи:*

1. Постановка проблеми.

Згенерувати всі  $k$ -елементні підмножини множини  $A$  з  $N$  чисел,  $A = \{1, 2, \dots, N\}$ .

2. Побудова моделі.

Є підмножина  $A$  яка складається з чисел від 1 до  $N$ , при цьому  $N$  повинно бути не менше 1. Є число  $k$  яке означає кількість елементів в підмножині.  $k$  повинно бути не менше 1 та не більше  $N$ .

3. Розроблення алгоритму.

Алгоритм генерації  $k$ -елементних підмножин множини  $A$  з  $N$  чисел,  $A = \{1, 2, \dots, N\}$ .

Вхідні дані:

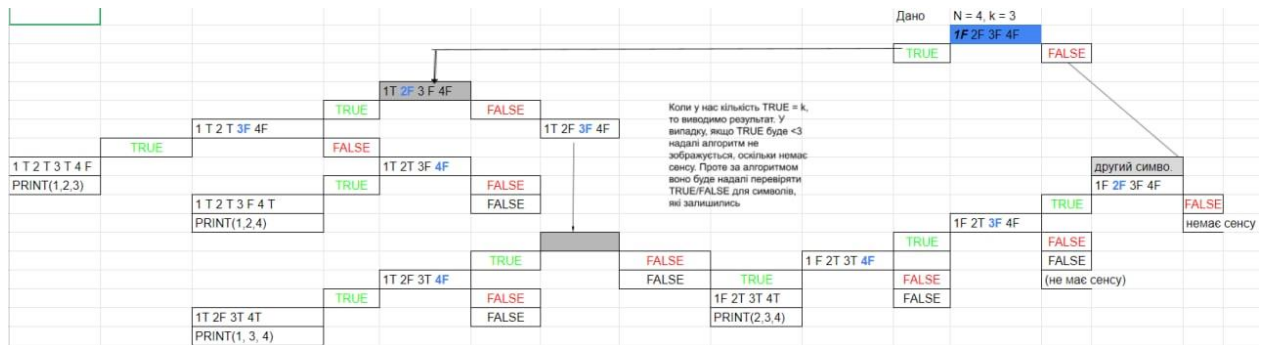
1.  $N$ - кількість елементів множини  $A$
2.  $k$  - кількість елементів у підмножинах.

Вихідні дані:

Усі можливі  $k$ -елементні підмножини

1. Генеруємо масив від 1 до  $N$  елементів.
2. Послідовно розглядаємо кожен елемент множини  $A$ . Рекурсія для двох випадків:
  - Якщо елемент включений, то поміщаємо елемент в підмножину, і збільшуємо доступний індекс;
  - Якщо елемент не включений в поточній підмножині, то пропускаємо елемент, не змінюючи індекс.
3. Коли підмножина досягає розміру  $k$ , друкуємо підмножину.
4. Повторити пункти 2,3 допоки не будуть виведені всі можливі підмножини

#### 4. Перевірка правильності алгоритму.



## 5. Реалізація алгоритму.

```
def subset(n, k):

    subset = list(range(k))

    if(n < 1 or k < 1 or k == n):
        print("Тут трабл")
        return

    array = list()
    for i in range(1, n + 1):
        array.append(i)

    subsetUtil(array, n, k,
               0, subset, 0)
```

## Основна функція

```
def subsetUtil(array, n, k, index, subset, i):
    if (index == k):
        for j in range(k):
            print(subset[j], end=" ")
        print(" ")
        return

    if (i >= n):
        return

    subset[index] = array[i]
    subsetUtil(array, n, k, index + 1, subset, i + 1)

    subsetUtil(array, n, k, index, subset, i + 1)
```

Допоміжна функція

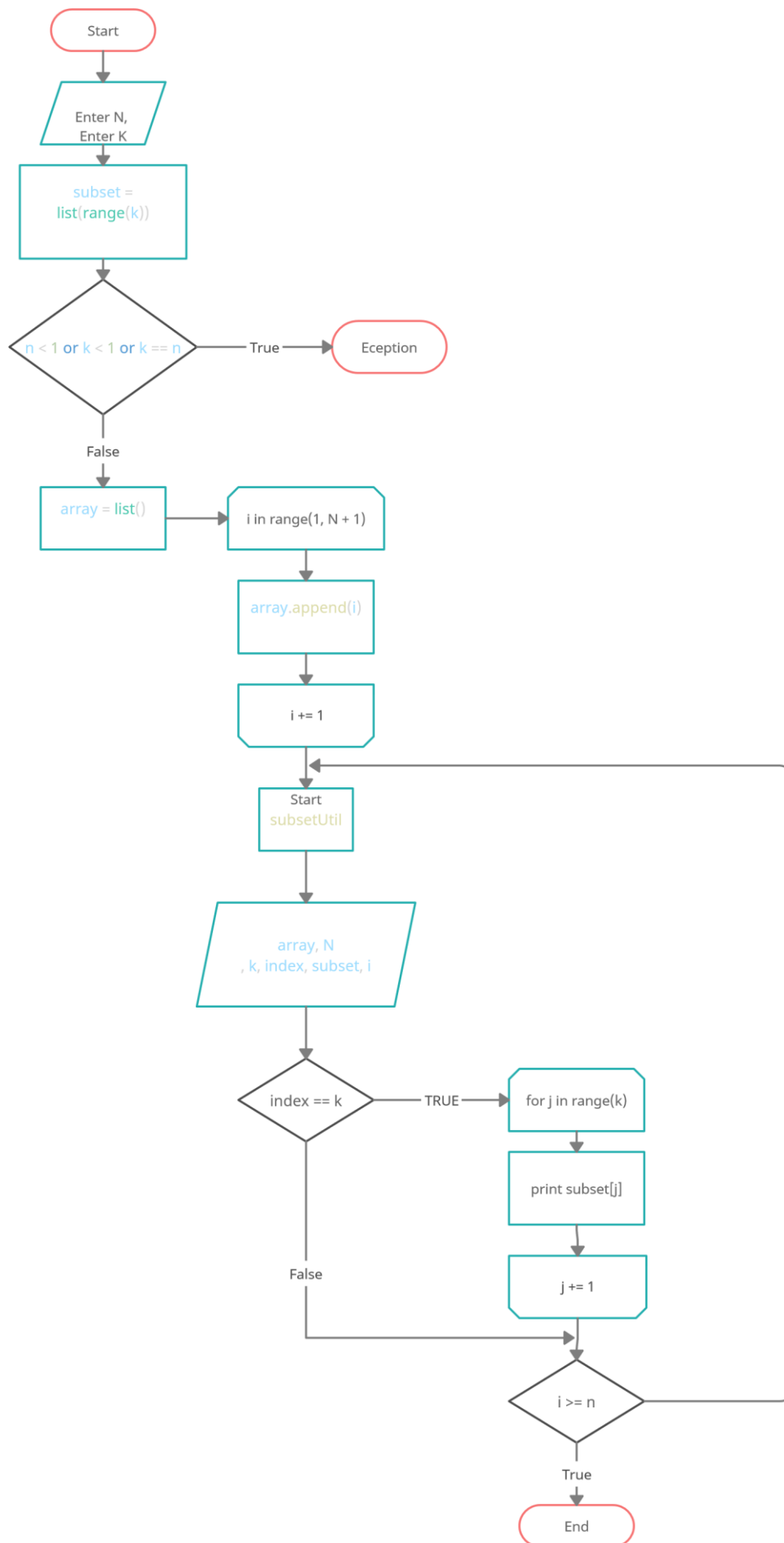
#### 6. Перевірка програми.

Підставимо значення з перевірки алгоритму. Значення  $k = 3$ ,  $N = 4$ .  
Перевіряємо повинно бути 4 підмножини.

```
k = 3 N = 4
1 2 3
1 2 4
1 3 4
2 3 4
```

Все вірно програма працює правильно.

#### 7. Складання документації (блок-схема алгоритму).



### *Висновки:*

На цій лабораторній роботі ми навчилися краще працювати в команді, а саме завдяки структурованій роботі(1. Постановка проблеми. 2. Побудова моделі. 3. Розроблення алгоритму. і так далі) ми легше розділили обов'язки та кожен сумлінно виконував свою задану окрему задачу, у випадках сумнівних ситуацій ми координувались між собою задля скорішого й ефективного вирішення проблеми. Навчилися створювати, описувати та реалізовувати алгоритми. Також більш ближче ознайомились з мовою Python.

[Посилання на репозиторій](#)

КИЇВ 2022