

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний аерокосмічний університет ім. М.Є. Жуковського
«Харківський авіаційний інститут»

Факультет радіоелектроніки, комп'ютерних систем та інфокомунікацій

Кафедра комп'ютерних систем, мереж і кібербезпеки (503)

Лабораторна робота № 6

з дисципліни	<u>Дослідження алгоритмів пошуку підрядка в рядку</u>
	(назва лабораторної роботи)
	<u>Моделі та структури даних</u>
	(шифр)

XAI.503.525a.03O.123-Комп'ютерна інженерія, ПЗ №9629619

Виконав студент гр.	525a	Литвиненко А.В.
29.11.2022	(№	(П.І.Б.)
	групи)	
(підпис, дата)		
Перевірив	канд. техн. наук, доцент	
	А. В. Шостак	
(підпис, дата)	(П.І.Б.)	

Харків – 2022

Тема роботи: дослідження алгоритмів пошуку підрядка в рядку.

Варіант 5

Задача 1

Частина 1. Постановка завдання

Умова:

Задание:

1. Разработать проект для исследования алгоритмов поиска подстроки в строке в соответствии с вариантом, приведенном в таблице 1.

2. Создать класс для исследования алгоритмов поиска подстроки в строке со следующими полями, свойствами и методами:

- образец размера M ,
- текст размера N ,
- метод задания образца (с параметрами алфавит и размер образца),
- метод задания текста (с параметрами алфавит и размер текста),
- метод реализации прямого поиска,
- метод реализации РК-поиска,
- метод получения хеша для РК-поиска,
- метод реализации КМП-поиска,
- метод для расчета префикс-функции (Z -функции),
- метод реализации БМ-поиска,
- метод для расчета таблицы стоп-символов,
- метод для расчета таблицы суфиксов,
- метод реализации Shift-And-поиска,
- метод для построения столбца $U(T(j))$,
- метод выполнения встроенного в $C\#$ поиска,
- результат поиска образца в тексте.

3. Разработать интерфейс проекта, *позволяющий*:

- задавать образец для поиска (шаблон) задавать строку;
- задавать строку (текст);
- осуществлять выбор алгоритма поиска в строке для исследования;
- осуществлять вывод информации о результатах поиска,
- осуществлять вывод информации о результатах исследования алгоритма поиска в строке (исходную строку, пошаговую работу алгоритма поиска в строке (при небольшой размерности строки), показатели качества работы алгоритма поиска в строке).

4. Создать проект, реализующую алгоритмы поиска в строке в соответствии с вариантом. В проекте *предусмотреть*:

- поиск заданного образца и индекса его первого вхождения;
- определение количества сравнений алгоритма поиска в строке Sr (или времени работы алгоритма поиска в строке).

5. При формировании строки предусмотреть возможность задания алфавита для строки и размера строки. При формировании образца

предусмотреть возможность задания алфавита для образца и размера образца.

6. Результатом работы алгоритма поиска в строке является номер первого вхождения образца в строку.

Умова з додатка:

Варианты задач по лабораторной работе:

1. Алгоритм прямого поиска в строке.
2. Алгоритм РК-поиска в строке.
3. Алгоритм КМП-поиска в строке.
6. Поиск с помощью встроенной функции языка C#.

Варианти:

5.	1, 2, 3, 6
----	------------

Основні вбудовані методи пошуку:

5.	FindIndex()	Array
----	-------------	-------

Частина 2. Схема класів

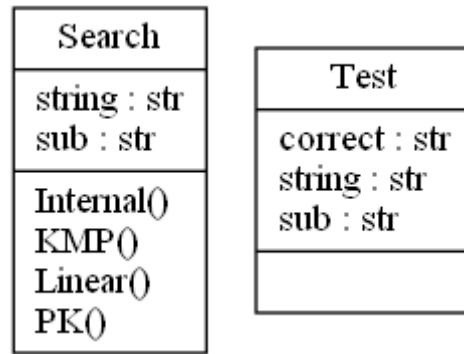


Рисунок 1 - схема класів

Частина 3. Текст програми

Відповідно до розробленого алгоритму в середовищі Microsoft Visual Studio була написана програма, яка наведена нижче.

Main.py

```
from time import time
```

```
class Search:
```

```
    def __init__(self, STRING: str = "", SUB: str = "") -> None:
        self.string = STRING
        self.sub = SUB
```

```
    def Linear(self):
```

```
        op = 0
        tmp = ""
        i = 0
```

```
        while i < len(self.string):
```

```
            if self.sub == tmp:
```

```
                return (True, op)
```

```
            if self.sub[0] == self.string[i]:
```

```
                j = 0
```

```
                tmp = ""
```

```
                while j < len(self.sub):
```

```
                    if self.sub[j] == self.string[i]:
```

```
                        tmp += self.sub[j]
```

```
                        j += 1
```

```
                        i += 1
```

```
                        op += 1
```

```
                    else:
```

```
                        break
```

```
                i += 1
```

```
                op += 1
```

```
        return (False, op)
```

```
    def PK(self):
```

```
        op = 0
```

```
        pattern = self.sub
```

```
        text = self.string
```

```
        q = 2147483647
```

```
        d = 10
```

```
        m = len(pattern)
```

```
        n = len(text)
```

```

p = 0
t = 0
h = 1
i = 0
j = 0

for i in range(m-1):
    h = (h*d) % q
    op += 1

# Calculate hash value for pattern and text
for i in range(m):
    p = (d*p + ord(pattern[i])) % q
    t = (d*t + ord(text[i])) % q
    op += 1

# Find the match
for i in range(n-m+1):
    op += 1
    if p == t:
        for j in range(m):
            op += 1
            if text[i+j] != pattern[j]:
                break

        j += 1
        if j == m:
            # print("Pattern is found at position: " + str(i+1))
            return (True, op)

    if i < n-m:
        t = (d*(t-ord(text[i])*h) + ord(text[i+m])) % q

    if t < 0:
        t = t+q
return (False, op)

def KMP(self):
    op = 0
    pat = self.sub
    txt = self.string

    M = len(pat)
    N = len(txt)

```

```

# create lps[] that will hold the longest prefix suffix
# values for pattern
lps = [0]*M
j = 0 # index for pat[]

```

```

# Preprocess the pattern (calculate lps[] array)
computeLPSArray(pat, M, lps)

```

```

i = 0 # index for txt[]
while i < N:
    op += 1
    if pat[j] == txt[i]:
        i += 1
        j += 1

    if j == M:
        # print ("Found pattern at index", str(i-j))
        return (True, op)
        j = lps[j-1]

```

```

# mismatch after j matches
elif i < N and pat[j] != txt[i]:
    # Do not match lps[0..lps[j-1]] characters,
    # they will match anyway
    if j != 0:
        j = lps[j-1]
    else:
        i += 1
return (False, op)

```

```

def Internal(self):
    return (False, 0) if self.string.find(self.sub) == -1 else (True, 0)

```

```

def computeLPSArray(pat, M, lps):
    len = 0 # length of the previous longest prefix suffix

    lps[0] # lps[0] is always 0
    i = 1

    # the loop calculates lps[i] for i = 1 to M-1
    while i < M:
        if pat[i] == pat[len]:

```

```

    len += 1
    lps[i] = len
    i += 1
else:
    # This is tricky. Consider the example.
    # AAACAAAA and i = 7. The idea is similar
    # to search step.
    if len != 0:
        len = lps[len-1]

    # Also, note that we do not increment i here
else:
    lps[i] = 0
    i += 1

```

```

def Time(self, func):
    start = time()
    data = func(self)
    end = time() - start
    print("Runtime: ", round(end, 10))
    return data

```

```

def main():
    s = Search()

    strings = open("tests.txt").readlines()
    sub = strings[0]
    strings = strings[1:]
    s.sub = sub
    # N = 10, 50, 100, 150, 200, 250, 300, 400, 500

```

```

for Method in (Search.Linear, Search.PK, Search.KMP, Search.Internal):
    print("Current method:\t", Method.__name__, end="\n\n")
    for string in strings:
        s.string = string

        result = Time(s, Method)
        if result[0] == True:
            print("[+] Passed!")
            print("[+] Operation Count : ", result[1])
        else:

```



```

        print("[-] FAILED!!!")
        print("Expected (%s) got %s" % (True, result[0]))
    print()
    print("\n")

```

```

print("<<< SUBS >>>")
print("="*40)

```

```

# N = 10, 50, 100, 150, 200, 250, 300, 400, 500
subs = [x.replace("\n", "") for x in open("subs.txt").readlines()]
string = subs[0]
s.string = string
subs = subs[1:]

```

```

for Method in (Search.Linear, Search.PK, Search.KMP, Search.Internal):
    print("Current method:\t", Method.__name__, end="\n\n")
    for sub in subs:
        s.sub = sub

        result = Time(s, Method)
        if result[0] == True:
            print("[+] Passed!")
            print("[+] Operation Count : ", result[1])
        else:
            print("[-] FAILED!!!")
            print("Expected (%s) got %s" % (True, result[0]))
    print()
    print("\n")

```

```

main()

```

tests.txt

```

hello
ZNvhelloas
lYZsLNbfFjOFqCBhelloAgSQaMWLCrixNBAthoUpPDtdNKdzSu
bOHsDKdpnloMggoqNMomZhhelloDEhFNBxGDkqJWZTDqVFtoqMlbaXwBXuftX
yYduhlUBKvXevGaDqjhKcVucXEulgeQhSjITLLsh
OogDxyzNkFmHpiodLOkkFCihelloyyEtXooHRcaaPONrODEmcdIllicQcGCESwXp
vTrhoihRhdztTfetQIIUpbbRHQAuUKvvvXTXyXIHRgySuBgjeHEQJPsoKAtkldsEaC
VLqIUyXkmBdByrAHcBSmGc
SSNVSQXcGTFWQCFwokFhVmOWazQMThelloOtQQuvVeBONdLynQRQGIEd
XbOWKxQZJSVLfbPFvSZArZqqvSFivWxcDqjJTSTbpujiVexKupRwwOwfyHkjqs
JtgBySIFYakxAeJmnBeCrRfzZTvtLHrXyqhQOaOdNNBblsgPEkGFswCbOzTLsQ
RfWSkLLUpnYZqLhPMV

```

FcGBytDaSoRmhrCFVtDeNRZhpjixGblDAWAMDgKnBwEiLehelloVjBeUaJVRP
TYqpZPPMIMLfDAHTKQNlpqydkOZPlmdpeubhbkxtkldGrTTPMEMuKXwSamRL
ciQgGixsPzvdbnACwdbrYSwiStQzoDvbJONGBuBOiUykoNYrXinFpEQVcNZLI
WPMKWRUDekBpbdrpdbfxQqnkbPiRmhtqrkYhnJvTKwBsETrSssZYtkImxJxOult
BgKmUa
MhfLkiSCLVChelloyQTAvpELxlQhOUgXxTfNwRRTKheMXWpfmoxrfVjEwohM
SydPcvECRVbOerOAZknGPafWYQSAvXYwyVPbyUgoyCRjfvEDRxvezteNBJKM
OxovdRvwLRcMssyCxWFLKUhcFzmLhWwPoStABAJItddidExjiPmhRdtXBHITBZ
ITvlxSMUksMufYFBfWpcdgHWwWDTMDQirTqejBotSBHQaaZzlTcrWPDFOJoE
eWqQmpCoXKyExQDEiKyFnYioXUxaMbfaFwyWdHPpGbICafQftimFSgSEuwT
AUDFwwtZBUIwMffGVSRNfroEtqXbuEjdvPKBInOplhelloKKqgmiYDJghxtqMX
qNTPITNbCsmINOMfUYDyUPsyNGPqVXBDhUuEAqTiZgKbsuLLeVPfxVKNV
IwChjkRHeaeCSYsqstGyFQaqOgwCLGTvzUIOhCMnvfZObJjVxTosvpoFehgnIxL
NmXBWjvUuTJnnxYSDRDLVNowAORzZZMNVipyLAMERatqBrujJoZfFnXlrDr
qsRnPMwJSNqAhGvubiBsSpkbPWCMMYmmQyNeJLwVvWkXjNALdnDuaxQNF
RMLhOdTGJqLUZlGCPVksJvtKFLhReGWxgvCnqNFthUbvkyDPVrIvsuaBXIVNF
HtXvGiliVQWNHQMmptLDCQzinZZDAdMQwGFfuBmg
mkHoAGmJnabCQUUTHLuvumqiClaOvXAZMmnpRhFTIvhjdQtGwrgeIYOdWtan
lSOdUHQMyuZQkGYqhdbFPTrngLioVhQINVGBQKARAcXgclEMlkChEjZrREe
ycVYTYbpYcHBOfpDhNguFgCynkwbellojHtWrjzpIuarmaIZIePFVdfzIPkeHpPnc
uVbQuwtNsHpHFQdmFSCIgOJCRUizRwpNsZxoiFODqwqejJDPGCGGuuTtHIXO
mzxKYvaGkFUxfuizvTxemhssZmDIIoerpZlonRsQMDgHMLjSIsCDWxmcHsSowh
LbloSMJJoSwxJAHYtynfbWlvwGGPjFbHEjhXpsxvfdnuVDXVACkFifvphEdzVen
zfDnAfjIJtXOLMRUCwwloWhRzoTFzlgkbFqnXxfuJSstwPWWKAgiwVNCwWor
kLvVsvYHbZliidJwgXLRRXIvlvYwwfctjPxfltoXFCBIHJodUOuJXjuClScXWlSt
ygl

subs.txt

PPHTEYoNmGwCgFotEqhRdCwpiPIDnPMPrnoqUCNJbXUCUFzUHZGIPGyphM
PfbJBiiRIsvsEeaUBbDYCFZvZQPHuPvUntQzenJWvdmUsEdwjJuceiMGgHOnxn
QEugyiBjRkqWYZeJSJjBLWkTLzqDIFGKoAJTrMxftaRmBgKttVUMqGYFELS
xahppYGoSEBlNLdrHoacATAXfoneJzXHSiGzUqfodipFvLIRGhqNElhTkuPzRFSE
NtQVtxBhXFbgolPaFhuhENkWFAtSSIPLIJKfhDJibmsMGOUWVkarfjftqmSwSs
bNkIYOijVgbjewQshqpOWyjSyoZtphelloBbtCyeSsJluoFhpbfcMTOWzPjbYimzrZI
YVYIdksFuRcAkehAzTOZLjzJekavIxrQwKrAdtkJjsOXMQienGJeXMKjCVDfNcr
AUngZfMlrMzDBjTQdtbPzDTigPraWNxGAGWNLksqVROAtxDtyzhDBelNTTM
cgVyRnRLlhRcAfvaliiAQkKRKeTspIwfmPZrAJfOtUlacIcjhJHHOTixnBMZlfzivU
UGBGogTKZDmXzdbBckMDOaynLaaTNBrCLZnfTkOVGyfUYZeanSSADSfyBjn
KSwnZMtZhrJsnTbWjeRPKLUAWUFLnOfrJxgRYVTZOQqbjRmadjJGkVmaFdMJ
wLNJoTHVyhLwjlrWnnrlzYZGCZIoRkiRQXsqcmOjTknFmLbOawHGRVmSVwvt
DeitzFLtbFzSPfQfQEnFfMFyjrOmmuQMRZJeSGOhvjSZKXGOWkkppOqJFTvRZJ
NiOHXdQjqlPmteisdRgyOYbfvCyVClkKaetGVCCpjdcGdxFScrnUbmRZHWtZVeN
kvmaYJMNslgpaiIAoisIKjhMEJLyzVwlaboQcesjqsXciipdjYwGARTzszrSAarYN
PascMndEzPZeMkOVXdzxoQEiVgyFRiGeFKHcLsekWrmjJZZCmUIQVtCnDpwju
aKWdjSTSOHPPHTEYoNmGwCgFotEqhRdCwpiPIDnPMPrnoqUCNJbXUCUFzUH

zGIPGyphMPfBJbiiRIsvsEeaUBbDYCFZvZQPHuPvUntQzenJWvdmUsEdwjjUcei
MGgHOnxnQEUgyiBjRkqWYZeJSJjBLWkTLzqDIFGKoAJTrMxFtaRmBgKttVU
MqGYFELSxahppYGoSEBlnLdrHoacATAXfoneJzXHSiGzUqfodipFvLIRGhqNELh
TkuPzRFSENtQVtxBhXFbgolPaFhuhENkWFAtSSIPLIJKfhDJibmsMGOUWVvKAR
fjftqmSwSsbNkIYOijVgbjewQshqpOWyjSyoZtphelloBbtCyeSsJIuoFhpbfcMTOWz
PjbYimzrZIYVYIdksFuRcAkehAzTOZLjzJekavIxrQwKrAdtkJjsOXMQienGJeXM
KjCVDfNcrAUngZfMlrMzDBjTQdtbPzDTigPraWNxGAGWNLksqVROAtxDtyzh
DBelNTTMcgVyRnRLIhRcAfvaliiAQkKRKeTspIwfmPZrAJfOtUlaclcjhJHHOTix
nBMZlfzivUUGBGogTKZDmXzdbBckMDOaynLaaTNBrCLZnfTkOVGyfUYZean
SSADSfyBjnKSwnZMtZhrJsnTbWjeRPKLuAWUFLnOfrJxgRYVTZOQqbjRmadjJ
GkVmaFdMJwLNJoTHVyhlwjlrWnnrlzYZGCZIoRkiRQXsqcmOjTknFmLbOawH
GRVmSVwvtDeitzFLtbFzSPfQfQEnFfMFyjrOmmuQMRZJeSGOhvjSZKXGOwkk
ppOqJFTvRZJNiOHXdQjqlPmteisdRgyOYbfvCyVClkKaetGVCCpjdcGdxFScrnUb
mrZHWtZVeNkvmaYJMnSLgpaiIAoisIKjhmEJLyzVwIaboQcesjqsXciipdjYwGAR
TzszrSAarYNPascMndEzPZeMkOVXdxoQeiVgyFRiGeFKHcLsekWrmjJZZCmU
IQVtCnDpwjuaKWdjSTSOhHTEYONmgwCgFotEqhRdCwpiPIDnPMPrnoqUCNjb
XUCUFzUHzGIPGyphMPfBJbiiRIsvsEeaUBbDYCFZvZQPHuPvUntQzenJWvdm
UsEdwjjUceiMGgHOnxnQEUgyiBjRkqWYZeJSJjBLWkTLzqDIFGKoAJTrMxFta
RmBgKttVUMqGYFELSxahppYGoSEBlnLdrHoacATAXfoneJzXHSiGzUqfodipFv
LIRGhqNELhTkuPzRFSENtQVtxBhXFbgolPaFhuhENkWFAtSSIPLIJKfhDJibmsM
GOUWVvKARfjftqmSwSsbNkIYOijVgbjewQshqpOWyjSyoZtphelloBbtCyeSsJIuoF
hpbfcMTOWzPjbYimzrZIYVYIdksFuRcAkehAzTOZLjzJekavIxrQwKrAdtkJjsOX
MQienGJeXMKjCVDfNcrAUngZfMlrMzDBjTQdtbPzDTigPraWNxGAGWNLksq
VROAtxDtyzhDBelNTTMcgVyRnRLIhRcAfvaliiAQkKRKeTspIwfmPZrAJfOtUla
clcjhJHHOTixnBMZlfzivUUGBGogTKZDmXzdbBckMDOaynLaaTNBrCLZnfTkO
VGyfUYZeanSSADSfyBjnKSwnZMtZhrJsnTbWjeRPKLuAWUFLnOfrJxgRYVTZ
OQqbjRmadjJGkVmaFdMJwLNJoTHVyhlwjlrWnnrlzYZGCZIoRkiRQXsqcmOjTk
nFmLbOawHGRVmSVwvtDeitzFLtbFzSPfQfQEnFfMFyjrOmmuQMRZJeSGOhvj
SZKXGOwkkppOqJFTvRZJNiOHXdQjqlPmteisdRgyOYbfvCyVClkKaetGVCCpjdc
GdxFScrnUbmrZHWtZVeNkvmaYJMnSLgpaiIAoisIKjhmEJLyzVwIaboQcesjqsX
ciipdjYwGARTzszrSAarYNPascMndEzPZeMkOVXdxoQeiVgyFRiGeFKHcLsek
WrmjJZZCmUIQVtCnDpwjuaKWdjSTSOhPHTEYONmgwCgFotEqhRdCwpiPIDnP
MPRnoqUCNjbXUCUFzUHzGIPGyphMPfBJbiiRIsvsEeaUBbDYCFZvZQPHuPvU
ntQzenJWvdmUsEdwjjUceiMGgHOnxnQEUgyiBjRkqWYZeJSJjBLWkTLzqDIFG
KoAJTrMxFtaRmBgKttVUMqGYFELSxahppYGoSEBlnLdrHoacATAXfoneJzXHS
iGzUqfodipFvLIRGhqNELhTkuPzRFSENtQVtxBhXFbgolPaFhuhENkWFAtSSIPLI
KfhDJibmsMGOUWVvKARfjftqmSwSsbNkIYOijVgbjewQshqpOWyjSyoZtphelloBb
tCyeSsJIuoFhpbfcMTOWzPjbYimzrZIYVYIdksFuRcAkehAzTOZLjzJekavIxrQwK
rAdtkJjsOXMQienGJeXMKjCVDfNcrAUngZfMlrMzDBjTQdtbPzDTigPraWNxGA
GWNLksqVROAtxDtyzhDBelNTTMcgVyRnRLIhRcAfvaliiAQkKRKeTspIwfmP
ZrAJfOtUlaclcjhJHHOTixnBMZlfzivUUGBGogTKZDmXzdbBckMDOaynLaaTNB
rCLZnfTkOVGyfUYZeanSSADSfyBjnKSwnZMtZhrJsnTbWjeRPKLuAWUFLnOfr
JxgRYVTZOQqbjRmadjJGkVmaFdMJwLNJoTHVyhlwjlrWnnrlzYZGCZIoRkiRQ
XsqcmOjTknFmLbOawHGRVmSVwvtDeitzFLtbFzSPfQfQEnFfMFyjrOmmuQMR

ZJeSGOhvjSZKXGOwkkppOqJFTvRZJNiOHXdQjqlPmteisdRgyOYbfvCyVClkKa
etGVCCpjdcGdxFScrnUbmRZHWtZVeNkvmaYJMnSLgpaiIAoisIKjhmEJLyzVwIa
boQcesjqsXciipdjYwGARTzszrSAarYNPascMndEzPZeMkOVXdzxoQEiVgyFRiG
eFKHcLsekWrmjJZZCmUIQAAAAVtCnDpwjuaKWdjSTSOhPHTEYoNmgwCgFo
tEqhRdCwpiPIDnPMPRnoqUCNJbXUCUFzUHzGIPGyphMPfBJbiiRIsvsEeaUBbD
YCFZvZQPHuPvUntQzenJWvdmUsEdwjjUceiMGgHOnxnQEugyiBjRkqWYZeJSJ
jBLWkTLzqDIFGKoAJTrMxFtaRmBgKttVUMqGYFELSxahppYGoSEBlnLdrHoa
cATAXfoneJzXHSiGzUqfodipFvLIRGhqNElhTkuPzRFSEntQVtxBhXFbgolPaFhu
hENkWFAtSSIPLIJKfhDJibmsMGOUWVkaRfjftqmSwSsbNkIYOijVgbjewQshqp
OWyjSyoZtphelloBbtCyeSsJluoFhpbfcMTOWzPjbYimzrZIYVYIdksFuRcAkehAzT
OZLjzJekavIxrQwKrAdtkJjsOXMQienGJeXMKjCVDfNcrAUngZfMlrMzDBjTQdt
bPzDTigPraWNxGAGWNLksqVROAtxDtyzhDBelNTTMcgVyRnRLlhRcAfvalii
AQkKRKeTsplwfmPZrAJfOtUlaclchJHHOTixnBMZlfzivUUGBGogTKZDmXzdb
BckMDOaynLaaTNBrCLZnfTkOVGyfUYZeanSSADSfyBjnKSwnZMtZhrJsnTbWj
eRPKLuAWUFLnOfrJxgRYVTZOQqbjRmadjJGkVmaFdMJwLNJoTHVyhIwJrWn
nrIzYZGCZIoRkiRQXsqcmOjTknFmLbOawHGRVmSVwvtDeitzFLtbFzSPfQfQEn
FfMFyjrOmmuQMRZJeSGOhvjSZKXGOwkkppOqJFTvRZJNiOHXdQjqlPmteisdR
gyOYbfvCyVClkKaetGVCCpjdcGdxFScrnUbmRZHWtZVeNkvmaYJMnSLgpaiIAo
isIKjhmEJLyzVwIaboQcesjqsXciipdjYwGARTzszrSAarYNPascMndEzPZeMkOV
XdzxoQEiVgyFRiGeFKHcLsekWrmjJZZCmUIQVtCnDpwjuaKWdjS
AAAAVtCnDp
AAAAVtCnDpwjuaKWdjSTSOhPHTEYoNmgwCgFotEqhRdCwpiPID
AAAAVtCnDpwjuaKWdjSTSOhPHTEYoNmgwCgFotEqhRdCwpiPIDnPMPRnoq
UCNJbXUCUFzUHzGIPGyphMPfBJbiiRIsvsEeaUBbDY
AAAAVtCnDpwjuaKWdjSTSOhPHTEYoNmgwCgFotEqhRdCwpiPIDnPMPRnoq
UCNJbXUCUFzUHzGIPGyphMPfBJbiiRIsvsEeaUBbDYCFZvZQPHuPvUntQzenJ
WvdmUsEdwjjUceiMGgHOnxnQEugyiBj
AAAAVtCnDpwjuaKWdjSTSOhPHTEYoNmgwCgFotEqhRdCwpiPIDnPMPRnoq
UCNJbXUCUFzUHzGIPGyphMPfBJbiiRIsvsEeaUBbDYCFZvZQPHuPvUntQzenJ
WvdmUsEdwjjUceiMGgHOnxnQEugyiBjRkqWYZeJSJjBLWkTLzqDIFGKoAJTr
MxFtaRmBgKttVUMqGYFEL
AAAAVtCnDpwjuaKWdjSTSOhPHTEYoNmgwCgFotEqhRdCwpiPIDnPMPRnoq
UCNJbXUCUFzUHzGIPGyphMPfBJbiiRIsvsEeaUBbDYCFZvZQPHuPvUntQzenJ
WvdmUsEdwjjUceiMGgHOnxnQEugyiBjRkqWYZeJSJjBLWkTLzqDIFGKoAJTr
MxFtaRmBgKttVUMqGYFELSxahppYGoSEBlnLdrHoacATAXfoneJzXHSiGzUqf
odipFvLIRG
AAAAVtCnDpwjuaKWdjSTSOhPHTEYoNmgwCgFotEqhRdCwpiPIDnPMPRnoq
UCNJbXUCUFzUHzGIPGyphMPfBJbiiRIsvsEeaUBbDYCFZvZQPHuPvUntQzenJ
WvdmUsEdwjjUceiMGgHOnxnQEugyiBjRkqWYZeJSJjBLWkTLzqDIFGKoAJTr
MxFtaRmBgKttVUMqGYFELSxahppYGoSEBlnLdrHoacATAXfoneJzXHSiGzUqf
odipFvLIRGhqNElhTkuPzRFSEntQVtxBhXFbgolPaFhuhENkWFAtSSIPLIJK
AAAAVtCnDpwjuaKWdjSTSOhPHTEYoNmgwCgFotEqhRdCwpiPIDnPMPRnoq
UCNJbXUCUFzUHzGIPGyphMPfBJbiiRIsvsEeaUBbDYCFZvZQPHuPvUntQzenJ
WvdmUsEdwjjUceiMGgHOnxnQEugyiBjRkqWYZeJSJjBLWkTLzqDIFGKoAJTr

MxFtaRmBgKttVUMqGYFELSxahppYGoSEBlnLdrHoacATAXfoneJzXHSSiGzUqf
odipFvLIRGhqNElhTkuPzRFSEntQVtxBhXFbgolPaFhuhENkWFAtSSIPLIJKfhDji
bmsMGOUWVkaRfjftqmSwSsbNkIYOijVgbjewQshqpOWyjSyoZtphelloBbtCyeSs
JluoFhpbfcMTOWzPjbYimzrZIYVYIdk
AAAAVtCnDpwjuaKWdjSTSOhPHTEYoNmngwCgFotEqhRdCwpiPIDnPMPRnoq
UCNJbXUCUFzUHzGIPGyphMPfBJbiiRIsvsEeaUBbDYCFZvZQPHuPvUntQzenJ
WvdmUsEdwjjUceiMGgHOnxnQEUgyiBjRkqWYZeJSJjBLWkTLzqDIFGKoAJTr
MxFtaRmBgKttVUMqGYFELSxahppYGoSEBlnLdrHoacATAXfoneJzXHSSiGzUqf
odipFvLIRGhqNElhTkuPzRFSEntQVtxBhXFbgolPaFhuhENkWFAtSSIPLIJKfhDji
bmsMGOUWVkaRfjftqmSwSsbNkIYOijVgbjewQshqpOWyjSyoZtphelloBbtCyeSs
JluoFhpbfcMTOWzPjbYimzrZIYVYIdksFuRcAkehAzTOZLjzJekavIxrQwKrAdtkJj
sOXMQienGJeXMKjCVDfNcrAUngZfMlrMzDBjTQdtbPzDTigPraWNxGAGWNL
ksqVRO

Частина 4. Порівняння

0 – означає, що прямує до нуля

N – кількість елементів

T – час виконання (в секундах)

S – кількість операцій

Таблиця 1 – Алгоритм прямого пошуку в рядку, довжина прикладу M=5

N	10	50	100	150	200	250	300	400	500
T	0	0	0	0	0	0	0	0	0.001
S	9	21	27	29	35	52	17	47	154

Таблиця 2 – Алгоритм РК пошуку в рядку, довжина прикладу M=5

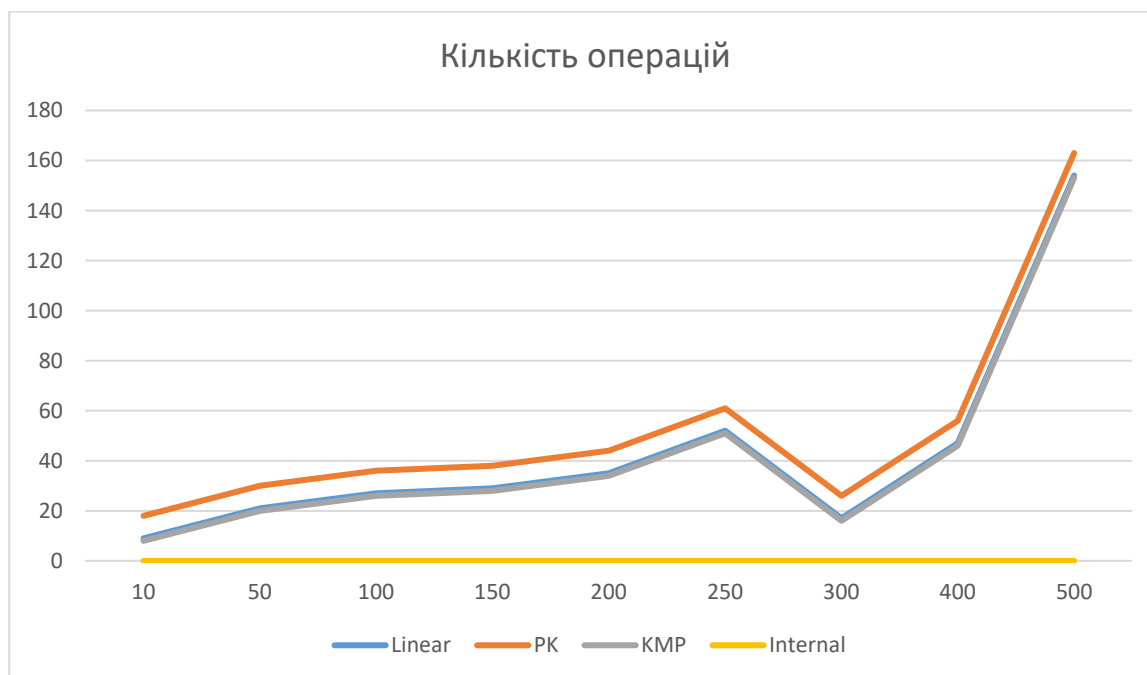
N	10	50	100	150	200	250	300	400	500
T	0	0	0	0	0	0	0	0	0.001
S	18	30	36	38	44	61	26	56	163

Таблиця 3 – Алгоритм КМР пошуку в рядку, довжина прикладу M=5

N	10	50	100	150	200	250	300	400	500
T	0	0	0	0	0	0	0	0	0.001
S	8	20	26	28	34	51	16	46	153

Таблиця 4 – Алгоритм internal пошуку в рядку, довжина прикладу M=5

N	10	50	100	150	200	250	300	400	500
T	0	0	0	0	0	0	0	0	0
S	0	0	0	0	0	0	0	0	0



```
Current method: Linear
Runtime: 0.0
[+] Passed!
[+] Operation Count : 9

Runtime: 0.0
[+] Passed!
[+] Operation Count : 21

Runtime: 0.0
[+] Passed!
[+] Operation Count : 27

Runtime: 0.0
[+] Passed!
[+] Operation Count : 29

Runtime: 0.0
[+] Passed!
[+] Operation Count : 35

Runtime: 0.0
[+] Passed!
[+] Operation Count : 52

Runtime: 0.0
[+] Passed!
[+] Operation Count : 17

Runtime: 0.0
[+] Passed!
[+] Operation Count : 47

Runtime: 0.0010032654
[+] Passed!
[+] Operation Count : 154
```

```
Current method: PK
Runtime: 0.0
[+] Passed!
[+] Operation Count : 18

Runtime: 0.0
[+] Passed!
[+] Operation Count : 30

Runtime: 0.0
[+] Passed!
[+] Operation Count : 36

Runtime: 0.0
[+] Passed!
[+] Operation Count : 38

Runtime: 0.0
[+] Passed!
[+] Operation Count : 44

Runtime: 0.0
[+] Passed!
[+] Operation Count : 61

Runtime: 0.0
[+] Passed!
[+] Operation Count : 26

Runtime: 0.0
[+] Passed!
[+] Operation Count : 56

Runtime: 0.0010025501
[+] Passed!
[+] Operation Count : 163
```

```
Current method: KMP
Runtime: 0.0
[+] Passed!
[+] Operation Count : 8

Runtime: 0.0
[+] Passed!
[+] Operation Count : 20

Runtime: 0.0
[+] Passed!
[+] Operation Count : 26

Runtime: 0.0
[+] Passed!
[+] Operation Count : 28

Runtime: 0.0
[+] Passed!
[+] Operation Count : 34

Runtime: 0.0
[+] Passed!
[+] Operation Count : 51

Runtime: 0.0
[+] Passed!
[+] Operation Count : 16

Runtime: 0.0
[+] Passed!
[+] Operation Count : 46

Runtime: 0.0010006428
[+] Passed!
[+] Operation Count : 153
```

```
Current method: Internal
Runtime: 0.0
[+] Passed!
[+] Operation Count : 0

Runtime: 0.0
[+] Passed!
[+] Operation Count : 0

Runtime: 0.0
[+] Passed!
[+] Operation Count : 0

Runtime: 0.0
[+] Passed!
[+] Operation Count : 0

Runtime: 0.0
[+] Passed!
[+] Operation Count : 0

Runtime: 0.0
[+] Passed!
[+] Operation Count : 0

Runtime: 0.0
[+] Passed!
[+] Operation Count : 0

Runtime: 0.0
[+] Passed!
[+] Operation Count : 0

Runtime: 0.0
[+] Passed!
[+] Operation Count : 0
```

Таблиця 5 – Алгоритм прямого пошуку в рядку, довжина рядку N=5000

M	10	50	100	150	200	250	300	400	500
T	0.003	0.002	0.002	0.0018	0.005	0.003	0.005	0.005	0.005
S	3992	4082	4032	4082	4132	4182	4232	4382	4482

Таблиця 6 – Алгоритм РК пошуку в рядку, довжина рядку N=5000

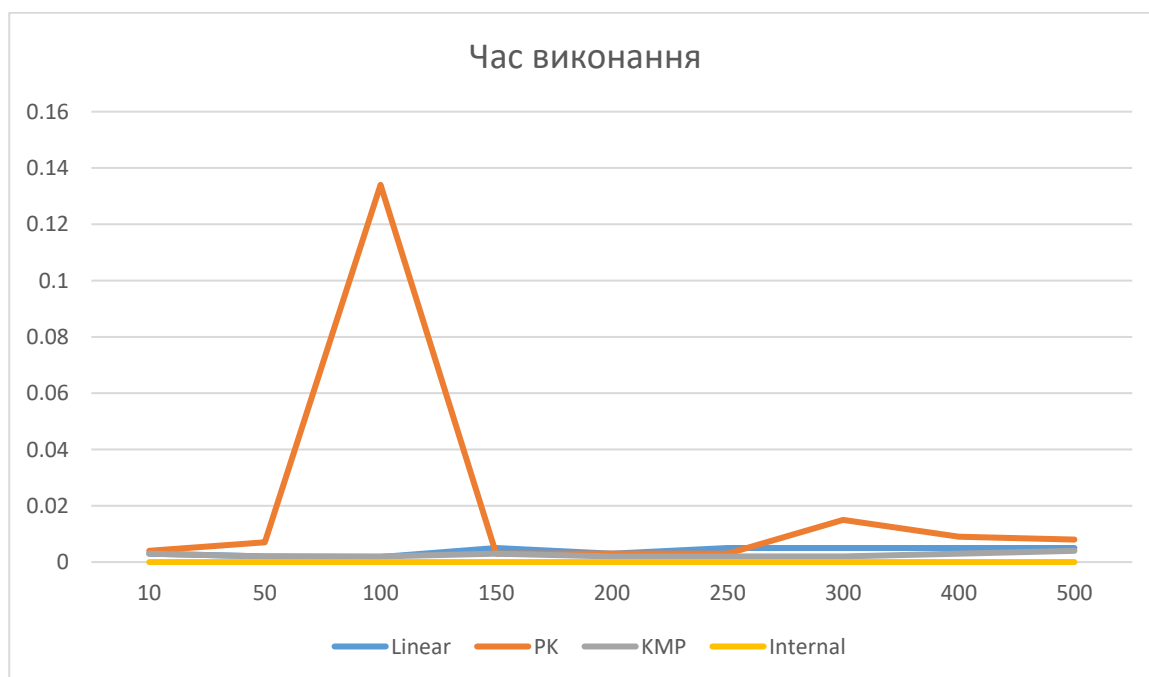
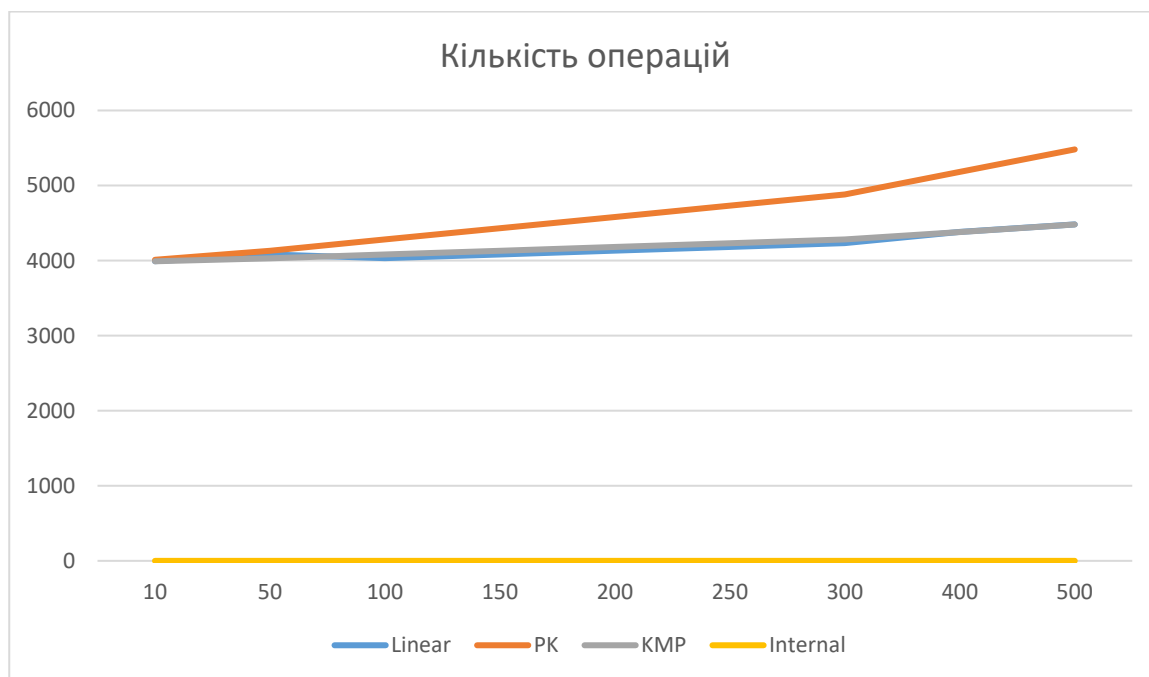
M	10	50	100	150	200	250	300	400	500
T	0.004	0.007	0.134	0.003	0.003	0.003	0.015	0.009	0.008
S	4011	4131	4281	4431	4581	4731	4881	5181	5481

Таблиця 7 – Алгоритм КМР пошуку в рядку, довжина рядку N=5000

M	10	50	100	150	200	250	300	400	500
T	0.003	0.002	0.002	0.003	0.002	0.002	0.002	0.003	0.004
S	3991	4031	4081	4131	4181	4231	4281	4381	4481

Таблиця 8 – Алгоритм internal пошуку в рядку, довжина рядку N=5000

M	10	50	100	150	200	250	300	400	500
T	0	0	0	0	0	0	0	0	0
S	0	0	0	0	0	0	0	0	0



Current method: Linear	Current method: PK	Current method: KMP	
Runtime: 0.0029988289 [+] Passed! [+] Operation Count : 3992	Runtime: 0.0388133526 [+] Passed! [+] Operation Count : 4011	Runtime: 0.0029895306 [+] Passed! [+] Operation Count : 3991	Current method: Internal
Runtime: 0.0020005703 [+] Passed! [+] Operation Count : 4032	Runtime: 0.0069956779 [+] Passed! [+] Operation Count : 4131	Runtime: 0.0020010471 [+] Passed! [+] Operation Count : 4031	Runtime: 0.0 [+] Passed! [+] Operation Count : 0
Runtime: 0.0014379025 [+] Passed! [+] Operation Count : 4082	Runtime: 0.0134384632 [+] Passed! [+] Operation Count : 4281	Runtime: 0.0019984245 [+] Passed! [+] Operation Count : 4081	Runtime: 0.0 [+] Passed! [+] Operation Count : 0
Runtime: 0.0057687759 [+] Passed! [+] Operation Count : 4132	Runtime: 0.003667593 [+] Passed! [+] Operation Count : 4431	Runtime: 0.0031218529 [+] Passed! [+] Operation Count : 4131	Runtime: 0.0 [+] Passed! [+] Operation Count : 0
Runtime: 0.00400424 [+] Passed! [+] Operation Count : 4182	Runtime: 0.0039954185 [+] Passed! [+] Operation Count : 4581	Runtime: 0.0019979477 [+] Passed! [+] Operation Count : 4181	Runtime: 0.0 [+] Passed! [+] Operation Count : 0
Runtime: 0.0014286041 [+] Passed! [+] Operation Count : 4232	Runtime: 0.0039832592 [+] Passed! [+] Operation Count : 4731	Runtime: 0.0027554035 [+] Passed! [+] Operation Count : 4231	Runtime: 0.0 [+] Passed! [+] Operation Count : 0
Runtime: 0.0022203922 [+] Passed! [+] Operation Count : 4282	Runtime: 0.0150051117 [+] Passed! [+] Operation Count : 4881	Runtime: 0.0020005703 [+] Passed! [+] Operation Count : 4281	Runtime: 0.0 [+] Passed! [+] Operation Count : 0
Runtime: 0.0049026012 [+] Passed! [+] Operation Count : 4382	Runtime: 0.009996891 [+] Passed! [+] Operation Count : 5181	Runtime: 0.0029346943 [+] Passed! [+] Operation Count : 4381	Runtime: 0.0 [+] Passed! [+] Operation Count : 0
Runtime: 0.0038757324 [+] Passed! [+] Operation Count : 4482	Runtime: 0.0079946518 [+] Passed! [+] Operation Count : 5481	Runtime: 0.003998518 [+] Passed! [+] Operation Count : 4481	Runtime: 0.0 [+] Passed! [+] Operation Count : 0

Висновки

Під час цієї лабораторної роботи я досліджував алгоритмів пошуку підрядка у рядку та реалізував проект для досліджування алгоритмів пошуку підрядка у рядку у зв'язку з варіантом.