String matching for same sized pattern and text

```python
import time
import random as rn
import matplotlib.pyplot as plt

def stringMatch(text1,pattern):
    if (len(text1)!= len(pattern)):
        raise Exception("Invalid Input!")
    a=0

    for i in range (len(pattern)):
        if(text1[i]==pattern[i]):
            a+=1
        else:
            break

    if (a==len(pattern)):
        print("The text and the pattern match")

    else:
        print("The text and the pattern dont match")

text=str(input("Please enter the text: "))
pattern=str(input("Please enter the pattern: "))
start=time.time()
stringMatch(text,pattern)
stop=time.time()

timeMatch=stop-start
```

```
Please enter the text: 1234567890
Please enter the pattern: 1234567890
The text and the pattern match
```

```python
text=str(input("Please enter the text: "))
pattern=str(input("Please enter the pattern: "))

start=time.time()
stringMatch(text,pattern)
stop=time.time()

timeMismatchBest=stop-start
```

```
Please enter the text: 1234567890
Please enter the pattern: 9876543210
The text and the pattern dont match
```

```python
text=str(input("Please enter the text: "))
pattern=str(input("Please enter the pattern: "))
```

```
start=time.time()
stringMatch(text,pattern)
stop=time.time()
```

```
timeMismatchWorst=stop-start
```

```
Please enter the text: 1234567890
Please enter the pattern: 1234567899
The text and the pattern dont match
```

```
print(timeMismatchWorst,timeMismatchBest,timeMatch)
```

```
0.00032806396484375 0.0003333091735839844 0.00042366981506347656
```

```
import numpy as np
```

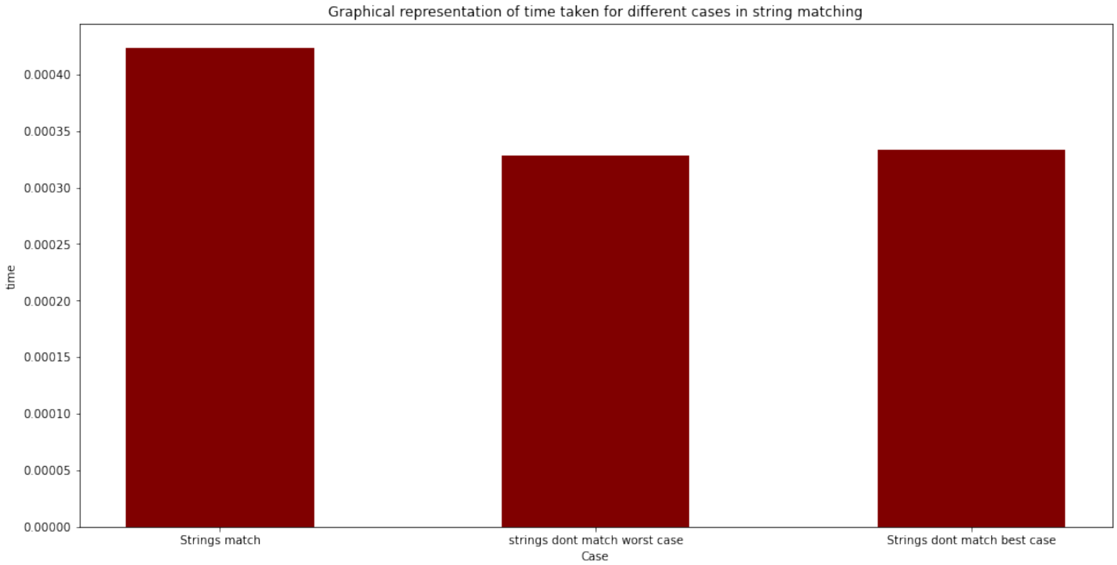
```
data = {'Strings match':timeMatch, 'strings dont match worst
case':timeMismatchWorst, 'Strings dont match best
case':timeMismatchBest}
```

```
time = list(data.keys())
case = list(data.values())
```

```
fig = plt.figure(figsize = (16, 8))
```

```
# creating the bar plot
plt.bar(time,case, color ='maroon',width = 0.5)
```

```
plt.xlabel("Case")
plt.ylabel("time")
plt.title("Graphical representation of time taken for different cases
in string matching")
plt.show()
```



Time Complexity

Here the elements are compared one by one and hence the time complexity for checking if the same sized text and pattern are matching is O(n).

Naïve String matching

```python
def naiveStringMatch(text,pattern):

    b=0

    for i in range (len(text)-len(pattern)+1):
        a=0
        for j in range (len(pattern)):
            if(text[j+i]==pattern[j]):
                a+=1

            else:
                b+=1
                break

        if (a==len(pattern)):
            print("The pattern is present at index",i)

    if(b==len(text)-len(pattern)+1):
        print("The pattern is not present in the text")



text=str(input("Please enter the text: "))
pattern=str(input("Please enter the pattern: "))

naiveStringMatch(text,pattern)
```

```
Please enter the text: 987987987
Please enter the pattern: 987
The pattern is present at index 0
The pattern is present at index 3
The pattern is present at index 6
```

```python
text=str(input("Please enter the text: "))
pattern=str(input("Please enter the pattern: "))

naiveStringMatch(text,pattern)
```

```
Please enter the text: 987987
Please enter the pattern: 987987
The pattern is present at index 0
```

```
text=str(input("Please enter the text: "))
pattern=str(input("Please enter the pattern: "))

naiveStringMatch(text,pattern)
```

```
Please enter the text: 987987987
Please enter the pattern: 123
The pattern is not present in the text
```

```
import time
import random as rn

masterArray=[]
timeStringMatching=[]
size=[5,10,15,20,35,50,100,200,500,1000,2000,3000,4000,5000,6000,7000,
8000,9000,10000]

for i in range(len(size)):
    masterArray.append(str((rn.randint(10**(size[i]-4),10**(size[i])-
3))*10**3+932))

pattern=str(932)

for i in range (len(size)):

    start=time.time()
    naiveStringMatch(masterArray[i],pattern)
    stop=time.time()
    timeStringMatching.append(stop-start)
    print("----")
```

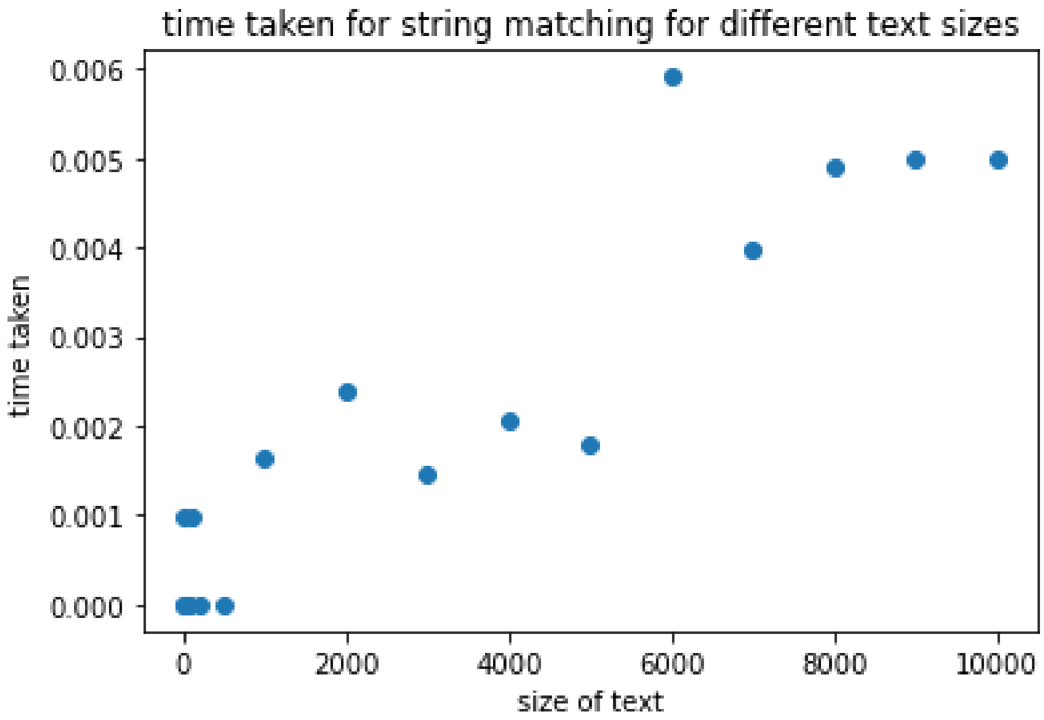
```
The pattern is present at index 5
----
The pattern is present at index 10
----
The pattern is present at index 15
----
The pattern is present at index 20
----
The pattern is present at index 35
----
The pattern is present at index 50
----
The pattern is present at index 100
----
The pattern is present at index 200
----
The pattern is present at index 165
The pattern is present at index 500
```

```
----
The pattern is present at index 18
The pattern is present at index 614
The pattern is present at index 694
The pattern is present at index 907
The pattern is present at index 1000
----
The pattern is present at index 1376
The pattern is present at index 2000
----
The pattern is present at index 284
The pattern is present at index 359
The pattern is present at index 3000
----
The pattern is present at index 788
The pattern is present at index 2296
The pattern is present at index 2336
The pattern is present at index 4000
----
The pattern is present at index 903
The pattern is present at index 907
The pattern is present at index 975
The pattern is present at index 3991
The pattern is present at index 4591
The pattern is present at index 5000
----
The pattern is present at index 449
The pattern is present at index 2518
The pattern is present at index 2747
The pattern is present at index 2983
The pattern is present at index 4108
The pattern is present at index 6000
----
The pattern is present at index 980
The pattern is present at index 989
The pattern is present at index 2544
The pattern is present at index 3867
The pattern is present at index 4049
The pattern is present at index 4379
The pattern is present at index 4770
The pattern is present at index 5768
The pattern is present at index 6300
The pattern is present at index 7000
----
The pattern is present at index 161
The pattern is present at index 537
The pattern is present at index 2061
The pattern is present at index 4258
The pattern is present at index 4677
The pattern is present at index 5209
The pattern is present at index 5328
The pattern is present at index 5407
The pattern is present at index 8000
----
The pattern is present at index 507
The pattern is present at index 733
The pattern is present at index 1342
The pattern is present at index 2258
The pattern is present at index 2446
The pattern is present at index 2489
The pattern is present at index 3139
The pattern is present at index 5916
The pattern is present at index 6713
The pattern is present at index 6866
The pattern is present at index 7025
The pattern is present at index 7464
The pattern is present at index 9000
----
The pattern is present at index 788
The pattern is present at index 1070
The pattern is present at index 1972
The pattern is present at index 3630
The pattern is present at index 4480
The pattern is present at index 5197
The pattern is present at index 5653
The pattern is present at index 6758
The pattern is present at index 6797
The pattern is present at index 6928
The pattern is present at index 7519
The pattern is present at index 8585
The pattern is present at index 9372
The pattern is present at index 9707
The pattern is present at index 10000
----
```

```
print(timeStringMatching)
```

```
[0.0009965896606445312, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0009732246398925781, 0.0, 0.0, 0.0016381740570068836,
0.002384185791015625, 0.0014734268188476562, 0.0020704269409179688,
0.0017988681793212890, 0.005910396575927734, 0.003986835479736328,
0.004911184310913086, 0.004984617233276367, 0.004983663558959961]
```

```
plt.scatter(size,timeStringMatching)
plt.xlabel('size of text')
plt.ylabel('time taken')
plt.title('time taken for string matching for different text sizes')
plt.show()
```



Time Complexity

In this case we have to check the pattern at every index one by one. The time taken to navigate through the text element by element is O(n) and the time taken to check if the pattern exists at that index is of the order O(m), where n is the length of the text and m is the length of the pattern. Hence the time complexity for naive string matching is O(n) * O(m) or simply O(m*n).

New naive string matching (treating pattern and text as numbers)

```
def newStringMatching (text,pattern):

    pattern1=int(pattern)
    for i in range (len(text)-len(pattern)+1):
        a=0
        b=0
        num=int(text[i])
        for j in range (i,i+len(pattern)-1):
            subtext=0
            num1=int(text[j+1])
            num=(10*num)+num1
            subtext=num

        if (subtext%13==pattern1%13):
            for k in range (len(pattern)):
                if(text[k+i]==pattern[k]):
                    a+=1

                else:
                    b+=1
                    break

                if (a==len(pattern)):
                    print("The pattern is present at index ",i)
```

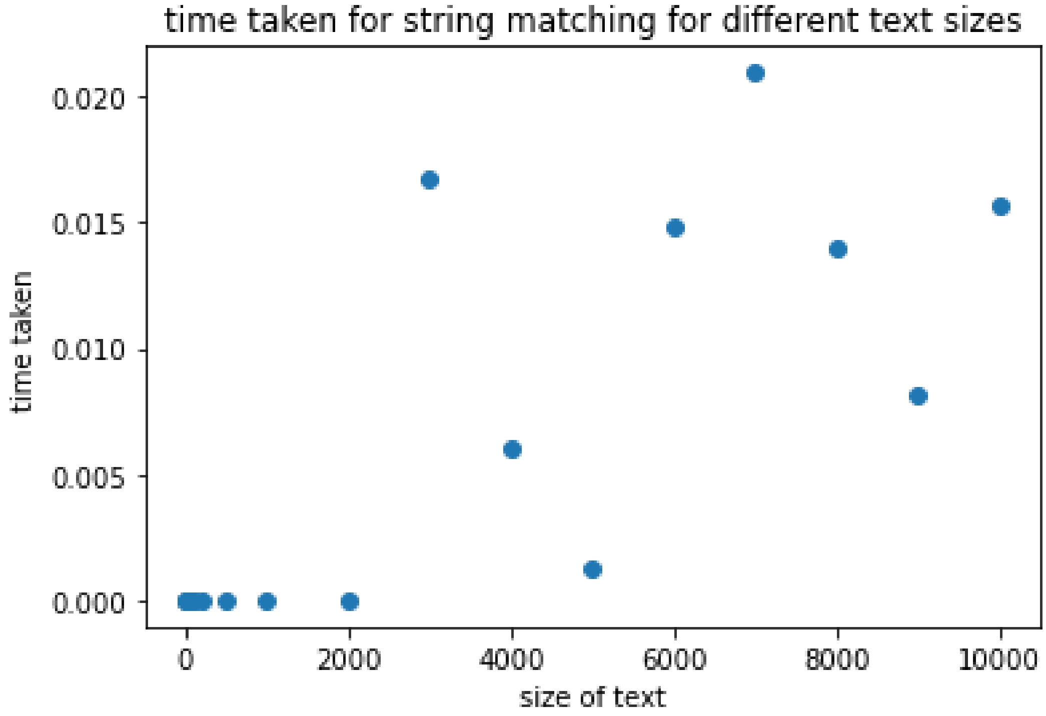
```
timeNewNaiveStringMatching=[]

import time
for i in range (len(size)):

    start=time.time()
    newStringMatching(masterArray[i],pattern)
    stop=time.time()
    timeNewNaiveStringMatching.append(stop-start)
    print("----")
```

```
----
----
----
----
----
----
----
----
----
The pattern is present at index  546
----
The pattern is present at index  139
The pattern is present at index  364
The pattern is present at index  662
The pattern is present at index  737
The pattern is present at index  1617
The pattern is present at index  1715
----
The pattern is present at index  765
----
The pattern is present at index  936
The pattern is present at index  2677
The pattern is present at index  2725
----
The pattern is present at index  283
The pattern is present at index  1656
The pattern is present at index  4090
The pattern is present at index  4554
The pattern is present at index  4644
----
The pattern is present at index  573
The pattern is present at index  835
The pattern is present at index  2242
The pattern is present at index  2697
The pattern is present at index  4540
The pattern is present at index  4804
----
The pattern is present at index  288
The pattern is present at index  2685
The pattern is present at index  4218
The pattern is present at index  4263
The pattern is present at index  4857
The pattern is present at index  5694
The pattern is present at index  6775
----
The pattern is present at index  356
The pattern is present at index  4532
The pattern is present at index  4751
The pattern is present at index  6624
The pattern is present at index  7143
The pattern is present at index  7564
The pattern is present at index  7880
----
The pattern is present at index  0
The pattern is present at index  1151
The pattern is present at index  3718
The pattern is present at index  3981
The pattern is present at index  4014
The pattern is present at index  5441
The pattern is present at index  5449
The pattern is present at index  8117
The pattern is present at index  8637
The pattern is present at index  8899
----
The pattern is present at index  27
The pattern is present at index  916
The pattern is present at index  2924
The pattern is present at index  3847
The pattern is present at index  4906
The pattern is present at index  5395
The pattern is present at index  5688
The pattern is present at index  5796
The pattern is present at index  7492
The pattern is present at index  9200
----
```

```
print(timeNewNaiveStringMatching)
```

```
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.016738176345825195, 0.006048440933227539, 0.0012364387512207031,
0.014863014221191406, 0.02093815803527832, 0.013956546783447266,
0.008120775222777832, 0.01569366455078125]
```

```
plt.scatter(size,timeNewNaiveStringMatching)
plt.xlabel('size of text')
plt.ylabel('time taken')
plt.title('time taken for string matching for different text sizes')
plt.show()
```



Optimized string matching

```
def optimizedStringMatching (text,pattern):

    pattern1=int(pattern)
    for i in range (len(text)-len(pattern)+1):

        num=int(text[i])
        for j in range (i,i+len(pattern)-1):
            subtext=0
            num1=int(text[j+1])
            num=(10*num)+num1
            subtext=num

        if (subtext==pattern1):
            print("The pattern exists at index ",i)
```

```
text='987987987'
pattern='987'
```

```
optimizedStringMatching(text,pattern)
```

```
The pattern exists at index  0
The pattern exists at index  3
The pattern exists at index  6
```

```
timeOptimizedNaiveStringMatching=[]
```

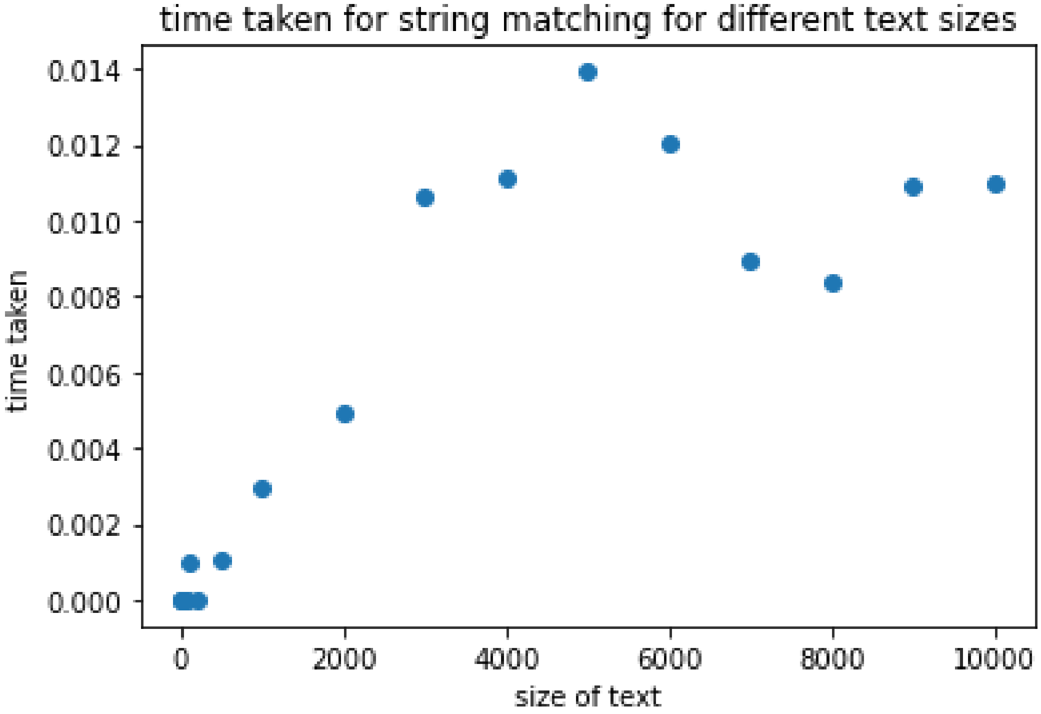
```
import time
for i in range (len(size)):

    start=time.time()
    optimizedStringMatching(masterArray[i],pattern)
    stop=time.time()
    timeOptimizedNaiveStringMatching.append(stop-start)
    print("----")
```

```
----
----
----
----
----
----
----
----
----
The pattern exists at index  546
----
The pattern exists at index  139
The pattern exists at index  364
The pattern exists at index  662
The pattern exists at index  737
The pattern exists at index  1617
The pattern exists at index  1715
----
The pattern exists at index  765
----
The pattern exists at index  936
The pattern exists at index  2677
The pattern exists at index  2725
----
The pattern exists at index  283
The pattern exists at index  1656
The pattern exists at index  4090
The pattern exists at index  4554
The pattern exists at index  4644
----
The pattern exists at index  573
The pattern exists at index  835
The pattern exists at index  2242
The pattern exists at index  2697
The pattern exists at index  4540
The pattern exists at index  4804
----
The pattern exists at index  288
The pattern exists at index  2685
The pattern exists at index  4218
The pattern exists at index  4263
The pattern exists at index  4857
The pattern exists at index  5694
The pattern exists at index  6775
----
The pattern exists at index  356
The pattern exists at index  4532
The pattern exists at index  4751
The pattern exists at index  6624
The pattern exists at index  7143
The pattern exists at index  7564
The pattern exists at index  7880
----
The pattern exists at index  0
The pattern exists at index  1151
The pattern exists at index  3718
The pattern exists at index  3981
The pattern exists at index  4014
The pattern exists at index  5441
The pattern exists at index  5449
The pattern exists at index  8117
The pattern exists at index  8637
The pattern exists at index  8899
----
The pattern exists at index  27
The pattern exists at index  916
The pattern exists at index  2924
The pattern exists at index  3847
The pattern exists at index  4906
The pattern exists at index  5395
The pattern exists at index  5688
The pattern exists at index  5796
The pattern exists at index  7492
The pattern exists at index  9200
----
```

```
print(timeOptimizedNaiveStringMatching)
```

```
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0009565353393554688, 0.0,
0.0010464191436767578, 0.002981424331665039, 0.004902839660644531,
0.010629892349243164, 0.011128425598144531, 0.013921022415161133,
0.012057781219482422, 0.008957862854003906, 0.008345603942871094,
0.010924577713012695, 0.010968685150146484]
```

```
plt.scatter(size,timeOptimizedNaiveStringMatching)
plt.xlabel('size of text')
plt.ylabel('time taken')
plt.title('time taken for string matching for different text sizes')
plt.show()
```


time taken for string matching for different text sizes
time taken
size of text