FIND AND REPLACE

Algorithm and Problem Solving Project

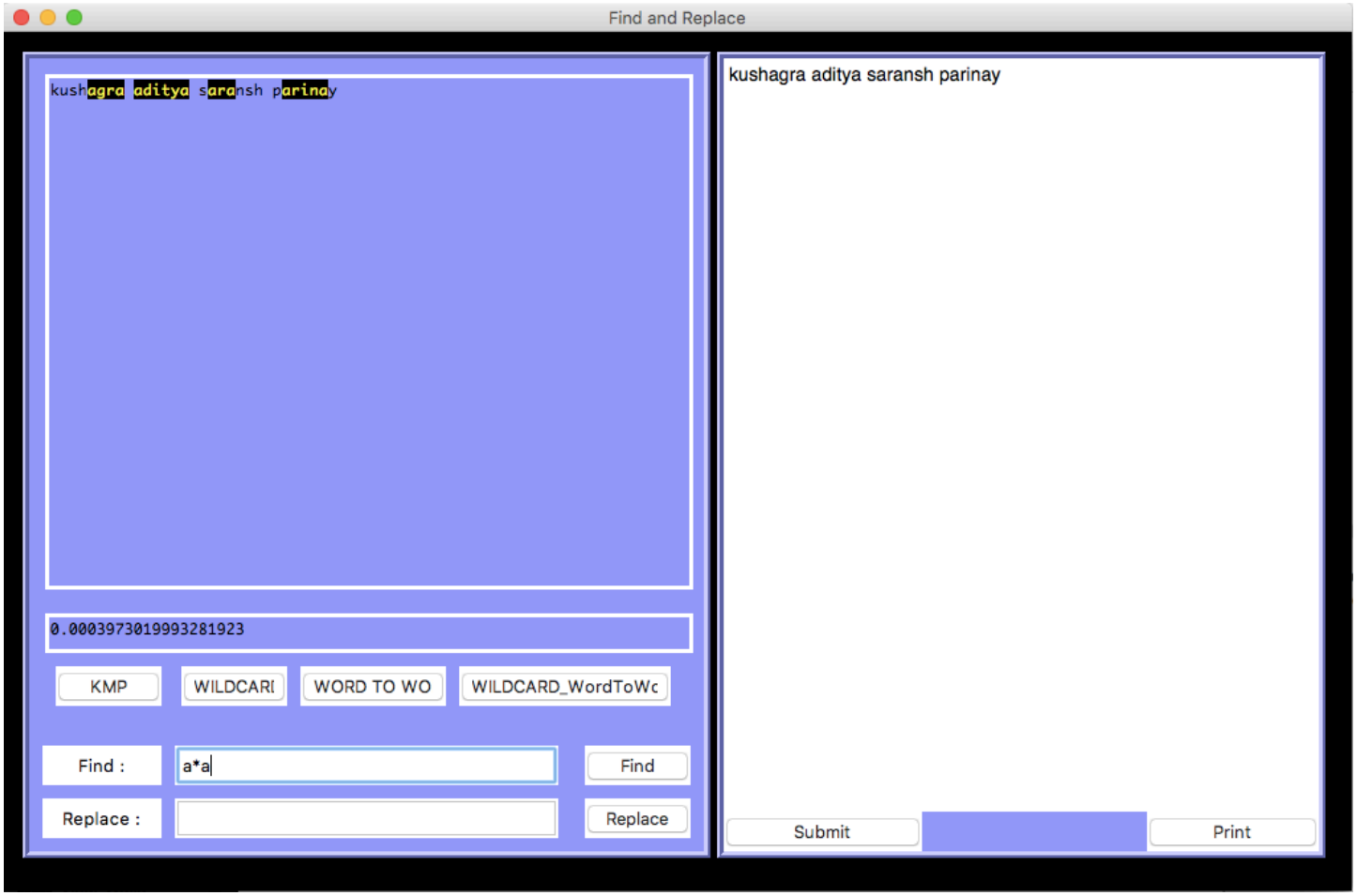
Features & Working

Sometimes we need to make our work easy by finding smaller text from a paragraph. Maybe you want to search your roll number from a list of thousand unsorted students. Doing it manually would take a lot of time. So this program help us to search various strings, exact words, wild-card expressions to make our work easier and faster.
What if you have to edit a particular data? Like there is an error in the spelling of your name, hence you want to edit it. So there is a replace command which is optimised to do the work faster.
It also shows the time in seconds that the single process takes.
The project is GUI based, which is made using tkinter framework of python. The text area takes the input of the text while the tabs on the left side are used to do the respective work. Print tab prints the new text with highlighted area.

- The string searching used here is KMP which takes a time of O(n+m) where n and m is the string length of text and pattern respectively.
- The word to word matching is further optimised to complexity of O(n'+m) at average case, where n' is the size of longest word in the text. It has a worst case complexity of O(n+m) only.
- The wild card algorithm uses a time complexity of O(n) compared to dynamic programming approach of O(n*m)
- The word to word wild card matching is further optimised to O(k) where k is the longest word in the text.

Front-end

We have designed the GUI on python with find, replace, search, wild-card tabs.



Back-end

The back-end is also made through python and connectivity between front-end and back-end is also done by python.

algo.py - /Users/kushagra98/Downloads/algo.py (3.6.4)

```
rom tkinter import *
mport tkinter.ttk as ttk
mport timeit
lobal inss
=0
oot= Tk()
oot.title("String Matching")
oot.geometry("1020x650+20+20")
oot.configure(background="BLACK")

ef InsRead(event):
        global inss
        inss = textfield.get(1.0,END)

        f=open("adi.txt","w+")
        f.write(inss)
        f.close()

ef print1(event):
        global inss
        global k
        global m
        global i
        global time_taken
        text = Text(frame1,background="#8080ff")
        text1 = Text(frame1,background="#8080ff")
        text.place(x=12,y=12,relheight=1,relwidth=1,height=-210,width=-20)
        text1.place(x=12,y=420,relheight=1,relwidth=1,height=-570,width=-20)
        text.insert(INSERT, inss)
        text1.insert(INSERT, time_taken)
        arr=k
        q=0
        for i in arr:
                k=str(i)
                j=str(i+m[q])
                k="1."+k
                l="1."+j
                text.tag_add("start",k , l)
                text.tag_config("start", background="black", foreground="yellow")
                q+=1
```

Ln: 1 Col: 0

Source Code:

```
from tkinter import *
import tkinter.ttk as ttk
import timeit
global inss
i=0
root= Tk()
root.title("Find and Replace")
root.geometry("1020x650+20+20")
root.configure(background="BLACK")

def InsRead(event):
	global inss
	inss = textfield.get(1.0,END)

	f=open("adi.txt","w+")
	f.write(inss)
	f.close()

def print1(event):
	global inss
	global k
	global m
	global i
	global time_taken
	text = Text(frame1,background="#8080ff")
	text1 = Text(frame1,background="#8080ff")
	text.place(x=12,y=12,relheight=1,relwidth=1,height=-210,width=-20)
	text1.place(x=12,y=420,relheight=1,relwidth=1,height=-570,width=-20)
	text.insert(INSERT, inss)
	text1.insert(INSERT, time_taken)
	arr=k
	q=0
	for i in arr:
		k=str(i)
		j=str(i+m[q])
		k="1."+k
		l="1."+j
		text.tag_add("start",k , l)
		text.tag_config("start", background="black", foreground="yellow")
		q+=1

def readfind(event):
	global str2
	str2=e1.get()

def lowercase(event):
	global inss
	inss=inss.lower()
	global str2
```

```
        str2=str2.lower()
        print(str2)
        print(inss)


def createarray(str2):
        j=0
        l=len(str2)
        i=0
        arr=[]
        for i in range(l):
                if(i==0):
                        arr.append(0)

                else:
                        if(str2[i]==str2[j]):
                                j+=1
                                arr.append(j)
                        else:
                                j=0
                                if(str2[j]==str2[i]):
                                        j+=1
                                arr.append(j)
        return arr

def kmp_wordtoword(event):
        start= timeit.default_timer()
        global time_taken
        global inss
        global k
        k=[]
        global m
        m=[]
        global str2
        start= timeit.default_timer()
        sss=inss
        sss1=sss.replace("."," ").replace(","," ")
        str3=sss1.split(" ")
        l2=len(str2)
        c=0
        l=len(str3)
        q=0
        pos=0
        for j in range(l):
                str1=str3[j]
                l1=len(str1)
                if(j==0):
                        pos=pos+l1
                else:
                        pos=pos+l1+1
                q=0
                if(l1==l2):
```

```
                    for i in range(l1):
                        if(str1[i]==str2[q]):
                            q+=1
                            if(q==l2):
                                c+=1
                                k.append(pos-l1)
                                m.append(l2)
                        else:
                            break
        stop = timeit.default_timer()
        time_taken=stop-start
        print(time_taken)
        print(c)
        print(k)
        print(m)

def kmp(event):
        start= timeit.default_timer()
        global time_taken
        global inss
        global k
        k=[]
        global m
        m=[]
        global str2
        l2=len(str2)
        c=0
        str1=inss
        arr1=createarray(str2)
        l1=len(str1)
        j=0
        c=0

        for i in range(l1):
            if(str1[i]==str2[j]):
                j+=1
            else:
                j=arr1[j]
                if(str1[i]==str2[j]):
                    j+=1;
            if(j==l2):
                c+=1
                m.append(l2)
                k.append(i-l2+1)
                j=0


        stop = timeit.default_timer()
        time_taken=stop-start
        print(stop-start)
        print(c)
```

```
        print(arr1)

def kmp_replace(event):
        start= timeit.default_timer()
        global time_taken
        global inss
        global k
        k=[]
        global m
        m=[]
        global str2
        l2=len(str2)
        c=0
        str1=inss
        arr1=createarray(str2)
        l1=len(str1)
        j=0
        c=0

        for i in range(l1):
                if(str1[i]==str2[j]):
                        j+=1
                else:
                        j=arr1[j]
                        if(str1[i]==str2[j]):
                                j+=1;
                if(j==l2):
                        c+=1
                        m.append(l2)
                        k.append(i-l2+1)
                        j=0


        stop = timeit.default_timer()
        rep_str=e2.get()
        print(k)
        time_taken=stop-start
        z=0
        w=0
        for i in k:
                s1=str1[0:i+z]
                s2=rep_str
                s3=str1[i+z+m[w]:]
                str1=s1+s2+s3
                inss=s1+s2+s3
                m[w]=len(s2)
                k[w]=i+z
                if(len(str2)>len(rep_str)):
                        z-=(len(str2)-len(rep_str))
                else:
                        z+=(len(rep_str)-len(str2))
                print(inss)
```

```
                w+=1

        print(stop-start)
        print(c)

        print(arr1)

def wildcard_word(event):
        start = timeit.default_timer()
        global inss
        global str2
        global time_taken
        str3=inss.split(" ")
        l2=len(str2)
        c=0
        l=len(str3)
        j=0
        pos1=0
        v=[]
        arr=[]
        for i in range(l2):
                arr.append(0)
                v.append(False)
        for i in range(l2):
                if(str2[i]=='*'):
                        j=i
                else:
                        arr[i]=j
        j=0
        i=0
        c=0
        temp=0
        global k
        k=[]
        global m
        global time_taken
        m=[]
        flag=0
        pos=0
        d=0
        print (arr)
        for d in range(l):
                str1=str3[d]
                l1=len(str1)
                i=0
                j=0
                temp=0
                pos=0
                if(d==0):
                        pos1=pos1+l1
                else:
                        pos1=pos1+l1+1
```

```
                for z in range(l2):
                        v[z]=False
                while (True):
                        print(d,i,j,c)
                        if(i==l1):
                                break
                        if(str2[j]=='*'):
                                v[j]=True
                                temp=1
                                j+=1
                                flag=1

                        elif(str2[j]==str1[i] or str2[j]=='?'):

                                v[j]=True
                                pos+=1
                                flag=0
                                j+=1
                                i+=1

                        elif(str2[j]!=str1[i] and temp==1):
                                j=arr[j]+1
                                pos+=1
                                if(str2[j]==str1[i]):
                                        i+=1
                                        j+=1
                                else:
                                        i+=1

                        elif(str2[j]!=str1[i]):
                                i+=1
                                pos=0
                                j=0
                                break

                        if(v[l2-1]==True):
                                c+=1
                                j=0
                                k.append(pos1-l1)
                                m.append(pos)
                                v[l2-1]=False
                                break
        print(k)
        print(m)
        print(c)
        stop=timeit.default_timer()
        time_taken=stop-start



def wildcard(event):
        start = timeit.default_timer()
```

```
global inss
global str2
str1=inss
l2=len(str2)
c=0
l1=len(str1)
j=0
arr=[]
v=[]
for i in range(l2):
        arr.append(0)
        v.append(False)

for i in range(l2):
        if(str2[i]=='*'):
                j=i
        else:
                arr[i]=j

j=0
i=0
c=0
temp=0
global k
k=[]
global m
global time_taken
m=[]
flag=0
pos=0
print (arr)
print(v)

while (True):
        print(i,j,flag,temp,pos)
        if(i==l1):
                break

        if(str2[j]=='*'):
                v[j]=True
                temp=1
                j+=1
                flag=1

        elif(str2[j]==str1[i] or str2[j]=='?'):

                v[j]=True
                pos+=1
                flag=0
                j+=1
                i+=1
```

```
                elif(str2[j]!=str1[i] and temp==1):
                        j=arr[j]+1
                        pos+=1
                        if(str2[j]==str1[i]):
                                i+=1
                                j+=1
                        else:
                                i+=1

                elif(str2[j]!=str1[i]):
                        i+=1
                        pos=0
                        j=0        #this has to be taken care of

                if(v[l2-1]==True):
                        c+=1
                        j=0
                        flag=0
                        m.append(pos)
                        k.append(i-pos)
                        temp=0
                        pos=0
                        v[l2-1]=False
        stop = timeit.default_timer()
        print(c)
        time_taken=stop-start
        print(time_taken)
        print(k)
        print(m)


frame1= Frame(root, background="#8080ff", relief=RIDGE, borderwidth=5)
frame1.pack(side = LEFT)
frame2= Frame(root, background="#8080ff", relief=RIDGE, borderwidth=5)
frame2.pack(side = LEFT)

frame1.place(x=15, y=15, relwidth=1, relheight=1, height=-40, width=-500)
frame2.place(x=540, y=15, relwidth=1, relheight=1, height=-40, width=-560)

l1=Label(frame1, height=10,width=10,text="Find :")
l2=Label(frame1, height=10,width=10,text="Replace :")

e1=Entry(frame1)
e2=Entry(frame1)

l1.place(x=10,y=520,relheight=1,relwidth=1,height=-570,width=-420)
l2.place(x=10,y=560,relheight=1,relwidth=1,height=-570,width=-420)

e1.place(x=110,y=520,relheight=1,relwidth=1,height=-570,width=-220)
e2.place(x=110,y=560,relheight=1,relwidth=1,height=-570,width=-220)

b1=Button(frame1,height=10,width=10,text="Find")
```

```
b2=Button(frame1,height=10,width=10,text="Replace")
b3=Button(frame2,height=10,width=10,text="Submit")
b4=Button(frame2,height=10,width=10,text="Print")
b5=Button(frame1,height=10,width=10,text="KMP")
b6=Button(frame1,height=10,width=10,text="WILDCARD")
b7=Button(frame1,height=10,width=10,text="WORD TO WORD")
b8=Button(frame1,height=10,width=10,text="WILDCARD_WordToWord")
b9=Button(frame2,height=10,width=10,text="case insensitive")

b3.bind("<Button-1>", InsRead)
b4.bind("<Button-1>", print1)
b1.bind("<Button-1>", readfind)
b5.bind("<Button-1>", kmp)
b6.bind("<Button-1>", wildcard)
b7.bind("<Button-1>", kmp_wordtoword)
b8.bind("<Button-1>", wildcard_word)
b2.bind("<Button-1>", kmp_replace)
b9.bind("<Button-1>", lowercase)


b1.place(x=420,y=520,relheight=1,relwidth=1,height=-570,width=-430)
b2.place(x=420,y=560,relheight=1,relwidth=1,height=-570,width=-430)
b3.place(x=0,y=570,relheight=1,relwidth=1,height=-570,width=-300)
b4.place(x=320,y=570,relheight=1,relwidth=1,height=-570,width=-320)
b5.place(x=20,y=460,relheight=1,relwidth=1,height=-570,width=-430)
b6.place(x=115,y=460,relheight=1,relwidth=1,height=-570,width=-430)
b7.place(x=205,y=460,relheight=1,relwidth=1,height=-570,width=-400)
b8.place(x=325,y=460,relheight=1,relwidth=1,height=-570,width=-350)
b9.place(x=167,y=570,relheight=1,relwidth=1,height=-570,width=-320)

textfield= Text(frame2, height=530, width=420,font='helvetica 14')

textfield.place(x=0, y=0, relheight=1, relwidth=1, height=-30)

root.mainloop()
```