קארפ רבין - השוואה לגירסה "הבטוחה"

השוואה בין הגירסה הרגילה של קארפ רבין שראינו בכיתה find_matches_KR, לגירסה ה"בטוחה" (שמופיעה בקבצי התרגול) find_matches_KR_safe.

הגירסה ה"בטוחה", עוברת על כל אינדקס שחשוד להיות מופע של pattern ב text ומבצעת השוואה בזמן O(m) האם אכן מדובר במופע של pattern. כלומר, גירסה זו לא תחזיר False positives.

עבור text="a"*n ו-pattern מהצורה : "a"*m כאשר m<n כל אינדקס בין 0 ל (n-m) הינו אינדקס שמייצג מופע חוקי של pattern ולכן יוחזר ע"י האלגוריתם. בגירסה ה"בטוחה" עבור כל אחד מהאינדקסים הללו תתבצע השוואה בזמן O(m) בין המחרוזת באורך m שמתחילה באינדקס למחרוזת pattern .

כלומר זמן הריצה יהיה O(m(n-m)) וקלט זה באמת ישקף את זמן הריצה הזה.

נריץ את הקוד הבא:

```
##############################################################
####
#competition between unsafe and safe versions
import time
text = "a"*100000
print("text = 'a'*",len(text))
for pattern in ["a"*1000, "a"*10000, "a"*30000, "a"*40000, \
                "a"*50000, "a"*60000, "a"*70000, "a"*80000]:
     print("pattern = 'a'*",len(pattern))
     for f in [find_matches_KR, find_matches_KR_safe]:
          t0=time.clock()
          f(pattern, text)
          t1=time.clock()
          print (f.__name__, t1-t0)
     print("") #newline
```

תוצאות הריצה (המשך בעמוד הבא):

```
>>>
text = 'a'* 100000
pattern = 'a'* 1000
find_matches_KR 0.13723983220418132
find_matches_KR_safe 0.31677717395809524
```

```
pattern = 'a'* 10000
find_matches_KR 0.12173683751382086
find_matches_KR_safe 1.2145096724965727

pattern = 'a'* 30000
find_matches_KR 0.10749898255922963
find_matches_KR_safe 2.5129464650495774

pattern = 'a'* 40000
find_matches_KR 0.10952439035462191
find_matches_KR_safe 2.9291809907721778

pattern = 'a'* 50000
find_matches_KR 0.09483061140788873
find_matches_KR_safe 2.9869641264792364

pattern = 'a'* 60000
find_matches_KR 0.09822300170093179
find_matches_KR_safe 2.8927714139209577

pattern = 'a'* 70000
find_matches_KR 0.0853466028925709
find_matches_KR_safe 2.5377787304155

pattern = 'a'* 80000
find_matches_KR 0.08074552059612117
find_matches_KR_safe 1.933736283299723
```

נשים לב שיש שינוי קטן מאד בזמן הריצה של הגירסה הרגילה כתלות בגודל ה pattern: ככל שגודל ה pattern גדל, זמן הריצה יורד. זה נובע מכך שככל ש m גדל כמות המחרוזות בגודל m ב text קטנה. מצד שני , העובדה שזמן הריצה כמעט ולא משתנה נובעת מכך שהשוואת זוג finerprints לוקחת זמן קבוע.

לעומת זאת, זמן הריצה של find_matches_KR_safe גדל משמעותית ככל ש m גדל כי אמנם כמות המועמדים קטנה ככל ש m גדל (כמות המועמדים היא n-m) אך כל השוואה למועמד עולה O(m). הפונקציה שמתארת היטב את זמן הריצה של גירסה זו (ובאה לידי ביטוי היטב בקלט הספציפי הזה) הינה O(m(n-m)). פונקציה זו מקבלת ערך מקסימלי כאשר m=n/2, וזה גם בא לידי ביטוי בהרצות , כאשר m= 50000.

נבצע כעת השוואה של שתי הגירסאות עבור מחרוזת אקראית מעל a-z.
נריץ את הקוד הבא:

```
n=100000
m=10000
text = gen_str(n)
pattern = gen_str(m)
print("random str of len n=", n, " , random pattern of length m=",m)
for f in [find_matches_KR, find_matches_KR_safe]:
    t0=time.clock()
    indices =  f(pattern, text)
    t1=time.clock()
    print(indices)
    print (f.__name__, t1-t0)
```

ונקבל:

```
>>>
random str of len n= 100000  , random pattern of length m= 10000
[]
find_matches_KR 0.11511271585829962
[]
find_matches_KR_safe 0.11603601128480961
```

כלומר עבור מחרוזת אקראית זמני הריצה של הגירסה הרגילה ושל הגירסה הבטוחה כמעט זהים.
למעשה רשימת המועמדים ריקה ולכן לא מתבצעת השוואה מפורשת של אף מועמד ל pattern.