

מבני נתונים ואלגוריתמים – 88-280-02

String matching – 8 תרגיל

תאריך הגשה: 24/1/16

הוראות הגשה:

יש להגיש את התרגיל דרך האתר – submit.cs.biu.ac.il

יש לציין בתחילת הקובץ בהערה שם ות.ז.

יש להגיש קובץ יחיד בשם targil8_c.c (למי שמגיש ב-C) או targil8_cpp.cpp (למי שמגיש ב-C++).

תיאור המשימה:

עליכם לכתוב תוכנית המקבלת בשורת הפקודה מספר n ואחריו מחרוזת ACII שנקרא לה S . לדוגמא, אם לקבוץ הריצה שלכם קוראים a.out ומריצים את הפקודה:

```
>> a.out 5 aabAgFumy
```

אז $S=$ aabAgFumy- i $n=5$.

התוכנית תעבור על כל תתי המחרוזות של S באורך n ותדפיס את כל תתי המחרוזות המופיעות **בדיוק פעמיים**. אם לא קיימת תת מחרוזת כזו, לא תדפיסו כלום.

כל הדפסה כזו תהיה בשורה נפרדת. בין המחרוזות והמספרים יהיה רווח יחיד.

דוגמא של קלט:

```
>> a.out 3 the_rain_in_spain_stays_mainly_on_the_drain
```

הפלט (כמובן ללא מה שכתוב באפור):

```
n_s 16 10 // the_rain_in_spain_stays_mainly_on_the_drain
the 34 0 // the_rain_in_spain_stays_mainly_on_the_drain
he_ 35 1 // the_rain_in_spain_stays_mainly_on_the_drain
rai 39 4 // the_rain_in_spain_stays_mainly_on_the_drain
```

שימו לב שתתי המחרוזות "ain" ו-"in" לא יופיעו בפלט מכיוון שהן מופיעות יותר מפעמיים!

על מנת שסדר ההדפסה יהיה זהה עבור כולם, עליכם **למיין** (בסיבוכיות נורמלית...) את הפלט שלכם לפי המספר הראשון (בסדר עולה), כלומר לפי מיקום המופע השני של תת מחרוזת מסוימת.

לדוגמא- אם אתם עוברים על המחרוזות ובודקים כל תת מחרוזת, ואתם מקבלים את התוצאה הבאה:

```
the 34 0
he_ 35 1
rai 39 4
n_s 16 10
```

אז עליכם להדפיס את תתי המחרוזות בצורה ממויינת בסדר הבא:

```
n_s 16 10
the 34 0
he_ 35 1
rai 39 4
```

מימוש באמצעות רבין קרפ

- עליכם להיעזר בפונקציית Hash של מחרוזות, שנסמן HS המקיימת שניתן לחשב את $HS(a_1 a_2 \dots a_n)$ מ- $HS(a_0 a_1 \dots a_{n-1})$ בזמן $O(1)$.
- עליכם לממש טבלת Hash, אליה תמפו ערכי Hash של תתי מחרוזות באורך n של S. אתם רשאים לממש את טבלת ה-Hash בכל דרך שלמדתם, אך מומלץ לממש Cuckoo Hashing שהוא קל ונוח למימוש. כדאי להקצות כ-50% יותר תאים ממספר האיברים שמכניסים לטבלת ה-Hash.
- המלצה נוספת – פונקציית Hash מוצלחת היא $HS(a_1 a_2 \dots a_n) = (\sum a_i p^i \% P)$ כאשר p מספר ראשוני קטן ו-P מספר ראשוני גדול (לדוגמא, $p = 109, P = 999997$). * כיצד תחשבו את hs_{i+1} מ- hs_i ב- $O(1)$? שימו לב שהפונקציה pow (העלאה בחזקה) לא עובדת ב- $O(1)$!

דרישות סיבוכיות:

זמן (ללא ההדפסות): $O(len(S))$.

מקום: $O(len(S))$.

הערה: מותר להשתמש בספריות string.h, math.h ובחלקת vector ב-STL (למי שמגיש ב-C++).

תזכורת – אלגוריתם רבין-קרפ:

בהינתן מחרוזת T לחיפוש T ותבניות P_1, \dots, P_k בגודל n תווים, האלגוריתם מוצא את כל ההופעות של התבניות ב-T בזמן $O(length(T)+k)$. האלגוריתם משתמש בפונקציית hash מיוחדת HS המקיימת שניתן לחשב את $HS(a_1 a_2 \dots a_n)$ מ- $HS(a_0 a_1 \dots a_{n-1})$ ב- $O(1)$ פעולות. האלגוריתם בקצרה:
אם $n > length(T)$ אין פיתרון וסיימנו. אחרת:
בונים טבלת H hash בגודל k ולכל $1 \leq i \leq k$ מבצעים $H[HS(P_i)] = i$.
נסמן $hs_i = HS(T[i]T[i+1] \dots T[i+n-1])$.
נחשב את hs_0 . אם $H[hs_0]$ מוגדר ומצאנו התאמה ולאחר בדיקה ש- $P_{H[hs_0]}$ אכן מופיעה ב-T (במקום המצופה), נדפיס את $H[hs_0]$.
כעת, עובר $0 < i < length(T)$ נבצע: חשב את hs_i מ- hs_{i-1} . אם $H[hs_i]$ מוגדר ומצאנו התאמה ולאחר הבדיקה ש- $P_{H[hs_i]}$ אכן מופיעה ב-T, נדפיס את $H[hs_i]$.

דוגמאות קלט פלט:

```
>> a.out 4 AAAAAaaaaa
AAAA 1 0
aaaa 6 5
>> a.out 5 AAAAAaaaaa
>> a.out 4 SheSellsSeaShellsOnTheSeaShore
ells 13 4
heSe 20 1
SeaS 22 8
eaSh 23 9
```

```
>>> a.out 8 123456789101112
>>> a.out 2 123456789101112
11 12 11
12 13 0
>>> a.out 3 aaaAAaaaaAAaaa
aaA 7 1
aAA 8 2
AAA 9 3
AAa 10 4
Aaa 11 5
```