# ממן 14

אלגוריתמיקה 20290

23.08.2019

## 

[**ממן**](#_6wpwt0gra8n2) **14 1**

[שאלה](#_940e35q6racc) 1 2

[סעיף א](#_31ovr3cl7qcs) 2

[סעיף ב](#_jhpgy7nn1qeu) 2

[שאלה](#_ckz2mbr8gfkp) 2 3

[שאלה](#_ucazi0bjo5ko) 3 4

[סעיף א](#_j8yk9lti7ws5) 4

[סעיף ב](#_2fmecfiqjmt9) 4

[שאלה](#_gkaiznhqou6f) 4 5

[שאלה](#_2piostpwh28s) 5 6

[שאלה](#_rizxtp5p77lq) 6 7

## 

## שאלה 1

### סעיף א

כאשר אורך סדרת האינדקסים חסום, יש מספר סופי של אפשרויות, ולכן הבעיה כריעה.

ישנו מספר סופי של סדרות אינדקסים באורך 1, שהוא n (אורך סדרות המילים),  
ישנו מספר סופי של סדרות אינדקסים באורך 2, שהוא .

.

.

.

ישנו מספר סופי של סדרות אינדקסים באורך N, שהוא

כמובן שסכום המספרים הסופיים הנ"ל הוא מספר סופי, ולכן ניתן לכתוב אלגוריתם שצריך לעשות מספר סופי של בדיקות, ולכן הבעיה כריעה.

### 

### סעיף ב

מעמוד 214 בספר הלימוד, נראה כי בעיית התאמת המילים תהפוך NP-שלמה אם נחסום באמצעות N את אורך סדרת האינדקסים המבוקשת.

## שאלה 2

באופן אינטואיטיבי נראה כי ברור שגם בעיית הנחש החדשה אינה כריעה, שכן היא רק מוסיפה אילוצים על הבעיה הרגילה שאינה כריעה.

אראה כיצד ניתן להכריע את בעיית נחש הדומינו במחצית המישור הרגילה באמצעות בעיה זו בהנחה כי היא כריעה (כלומר קיים אורקל הפותר אותה):

עבור קלט T,W,V לבעיית נחש הדומינו במחצית המישור הרגילה,  
ניצור T+1 מרצפות, t1 כך שיש לה מכל צידיה צבע שאין באף מרצפת בT, בנוסף, לכל t מ- T נוסיף משבצת כך שפאה אחת מתאימה ל- t ופאה הצמודה לה מתאיה ל- t1, בצורה המאפשרת התאמה של t ל- t1 מכיוון יחיד:



ברור כי המשבצת t1 אינה מתאימה לאף משבצת אחרת ולכן בהכרח תהיה המשבצת הראשונה או האחרונה, ואחריה תבוא משבצת ההתאמה המתאימה.

נעביר את הקלטים (עם ti לכל אחת המרצפות החדשות) לבעיה המתוארת.

נראה כי קיים נחש בבעיה החדשה עם הקלט החדש אם ורק אם קיים נחש עבור הקלט המקורי לבעיה המקורית.

נניח כי קיים נחש בבעייה המקורית, אז ברור שניתן לצרף למרצפת הראשונה שלו את מרצפת ההתאמה המתאימה, ואלייה לצרף את המרצפת t1.

נניח כי קיים נחש בבעייה החדשה, אז כאמור המרצפת t1 היא בהכרח הראשונה או האחרונה, אז ניתן למחוק אותה ואת משבצת ההתאמה, וקיבלנו נחש שיוצא מהמיקום של V המקורי.

## שאלה 3

### סעיף א

בעיית הטוטליות שואלת האם תוכנית עוצרת על כל קלטיה החוקיים.

### סעיף ב

נניח כי בעיית הטוטליות כריעה, אז קיים אורקל הפותר אותה.

נראה כי אם בעיה זו כריעה, אז גם בעיית העצירה כריעה.

עבור קלט לבעיית העצירה תכנית A וקלט X, ניצור תכנית B אשר כל תכליתה להריץ את A על X.

תכנית B מתעלמת מהקלט שלה, ולכן תעצור (על כל קלטיה) אם ורק אם תכנית A תעצור על X.

נעביר לאורקל של בעיית הטוטליות את B.

מכך שבעיית העצירה כידוע אינה כריעה, ההנחה כי בעיית הטוטאליות כריעה אינה נכונה, ומתקבל כי הבעיה איננה כריעה.

## שאלה 4

בהנחה וקיים בידינו אורקל לבעיית הטוטליות, ניתן לבנות אלגוריתם אשר מקבל קלט מספר טבעי ועוצר רק כאשר הוא מוצא מספר ראשוני ז'רמן הגדול מהקלט.

אם האלגוריתם עוצר עבור כל קלט, אז לכל מספר טבעי קיים ראשוני ז'רמן גדול ממנו. כלומר, יש אינסוף מספרים ראשוניים-ז'רמן (כי יש אינסוף מספרים טבעיים כקלט).

האלג':

|  |
| --- |
| A(x):  while true:  if GermainPrime(x):  return true;  else:  x←x+1 |
| GermainPrime(x):  if (x is prime) and (2x+1 is prime): // using AKS primality test  return true;  return false; |

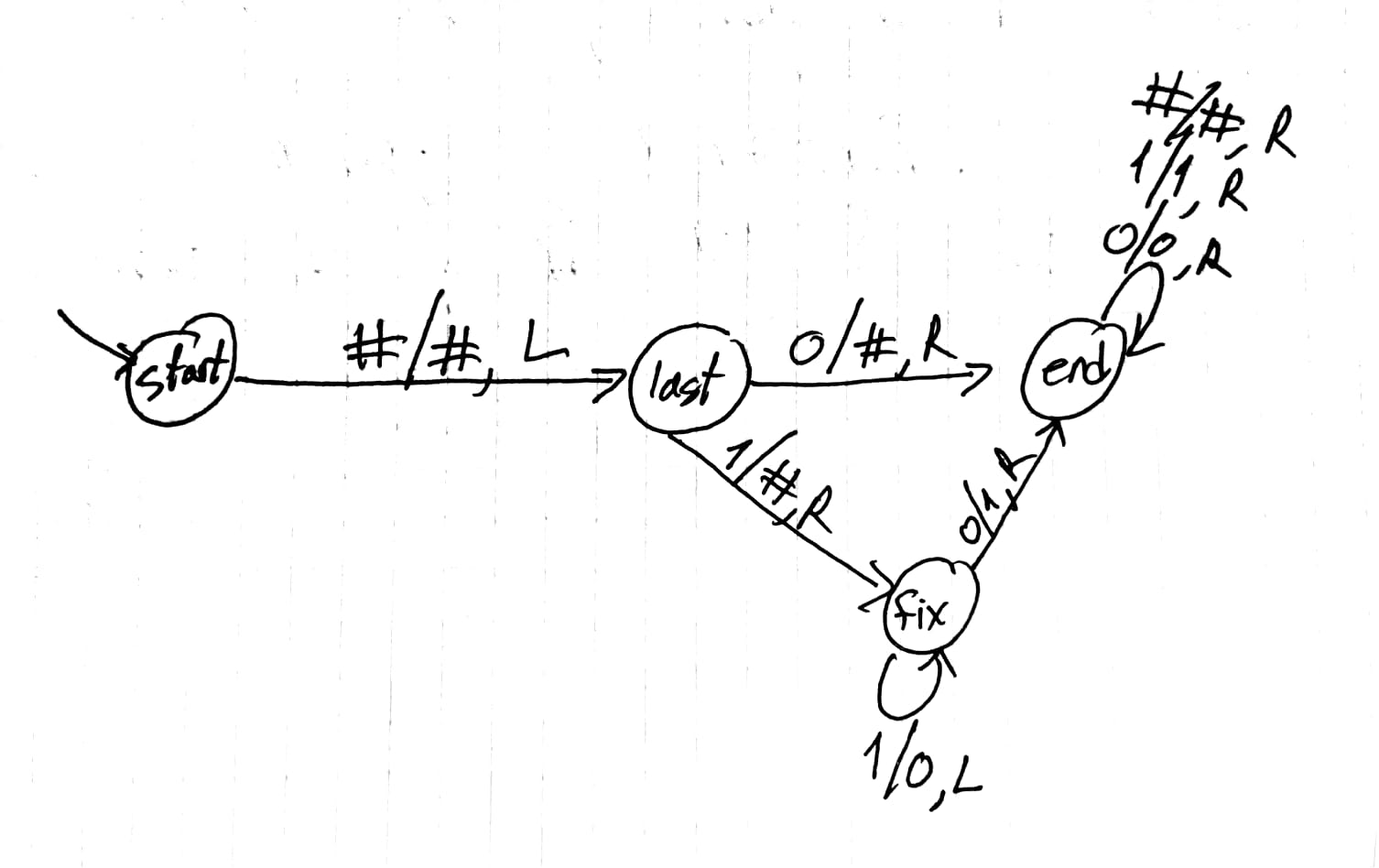
נעביר את A לאורקל ונקבל הכרעה לבעיה אם קיימים אינסוף מספרים שהם ראשונים ז'רמן.

## שאלה 5

הפונקציה מחלקת את המספר ב- 2 ומעגלת כלפי מעלה, בהסתכלות בינארית, אם המספר במיקום החלש (הימני ביותר) הוא 0 אז מדובר בחלוקה רגילה ב2 וניתן פשוט למחוק את מספר זה.

אם המספר במיקום החדש הוא 1, אז צריך למחוק אותו ולהוסיף 1, כלומר לכתוב 1 באפס הקרוב ביותר משמאל, תוך איפוס ה1ים שבדרך.

כמו כן המצב **end הוא מצב מקבל** (שכחתי לסמן זאת בשרטוט).



## שאלה 6

בהנחה שהמקרה הריק אינו מהווה התאמת מילים חוקית, למיטב הבנתי אין הכרח שהנתון יבטיח קיום של סדרה המהווה התאמת מילים חוקית.

נתבונן בדוגמה:

עבור סדרות מילים באורך 2 (מעל א"ב של אות אחת בלבד {a} ) הבאות:

X1=aaa, X2=a. Y1=a, Y2=aa.  
נראה כי עבור i=1, j=2 מתקיים כי וכי , אבל ברור כי לא קיימת התאמת מילים חוקית.