

## הרצאה 20: אלגוריתמים אקראיים

Textbook: Cormen, Leiserson, Rivest,  
Stein. Introduction to Algorithms.

מרצה: אורי שטמר

באלגוריתם דטרמניסטי – בכל צעד בריצה, המהלך של האלגוריתם נקבע באופן חד משמעי ע"י הקלט והצעדים הקודמים של האלגוריתם, ולכל קלט יש פלט יחיד. בהרצאות הקרובות נדבר על אלגוריתמים אקראיים. אלו אלגוריתמים אשר מטילים מטבעות במהלך הריצה ועפ"י תוצאת המטבע חליטים מה לעשות.

**אלגוריתם אקראי:** מטיל מטבעת. מטבע יוצא 0 בהסתברות  $\frac{1}{2}$  ויוצא 1 בהסתברות  $\frac{1}{2}$ . על פי תוצאת המטבע מחליטים מהו הצעד הבא של האלגוריתם.

### דוגמה פשוטה לאלגוריתם אקראי:

1. הטל מטבע
2. אם יצא 0 החזר 3
3. אם יצא 1 החזר 5

### נקודה חשובה:

- באלג' דטר' דרשנו – לכל קלט האלג' צריך להחזיר תשובה נכונה **תמיד**
- באלג' אקראי נדרוש – לכל קלט האלג' מחזיר תשובה נכונה **בהסתברות גבוהה**

דוגמה: מותר לטעות בהסתברות  $\frac{1}{2^{100}}$

מבחינה פילוסופית יש לנו כאן משהו מטריד. הרצנו אלגוריתם, קיבלנו תשובה (למשל "אין מסלול"), ואנחנו לא יכולים להסתמך עליה ב-100%. מבחינה מעשית, מאורע בהסתברות כמו  $\frac{1}{2^{100}}$  לא יקרה.

כדי להמחיש זאת: ההסתברות לזכות בלוטו היא בערך  $2^{-24}$ . ההסתברות של  $2^{-100}$  זה בערך ההסתברות שאשחק בלוטו 4 פעמים ובכולם אזכה מקום ראשון...

**המטרה:** לתכנן אלגוריתמים אקראיים יותר יעילים מאלגוריתמים דטרמניסטיים.

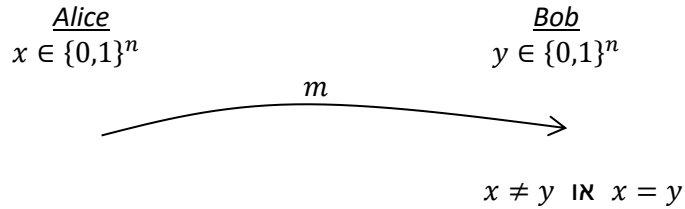
נתחיל מבעיה פשוטה:

### בעיית שוויון בין מחרוזות

יש לנו 2 משתתפים: אליס ובוב. לאליס יש קלט  $x \in \{0,1\}^n$  ולבוב יש קלט  $y \in \{0,1\}^n$ . כלומר לכל אחד משני המשתתפים יש קלט שהוא מחרוזת של  $n$  ביטים. מטרה: לבדוק האם  $x = y$ . תנאי המשחק: אליס שולחת לבוב הודעה אחת  $m$  ולאחר מכן בוב מכריז האם  $x = y$  או  $x \neq y$ .

מוטיבציה: להשוות בין שני דטה-בייסים.

בציור:



**פתרון פשוט:** אליס שולחת את  $x$  לבוב, ובוב בודק האם  $x = y$  או לא. תקשורת:  $n$  ביטים.

**האם אפשר לתכנן פרוטוקול בו אליס שולחת פחות מ-  $n$  ביטים?**

**טענה:** בכל פקוטורול דטרמיניסטי אליס שולחת לפחות  $n$  ביטים.

**הוכחה:**

נניח בשלילה שאליס שולחת לכל היותר  $n - 1$  ביטים.

ישנן  $2^{n-1}$  הודעות אפשריות של אליס.

לאליס יש  $2^n$  קלטים אפשריים.

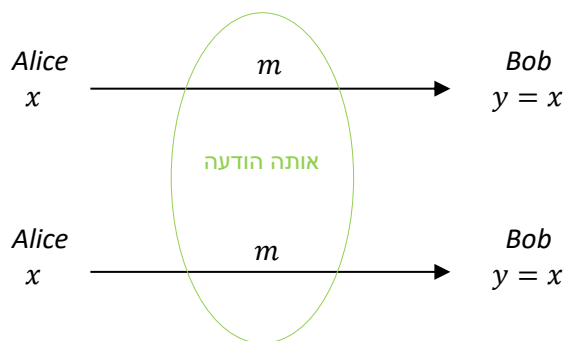
מעיקרון שוברך היונים, קיימים 2 קלטים  $x \neq x'$  שעליהם אליס שולחת את אותה ההודעה, שנסמנה ב-  $m$ .

מה בוב עונה כאשר הוא מחזיק בקלט  $y = x$  ומקבל הודעה  $m$ ?

- אם בוב עונה "שווים" אז הפרוטוקול טועה כאשר אליס החזיקה  $x'$

- אם בוב עונה "שונים" אז הפרוטוקול טועה כאשר אליס החזיקה  $x$

בציור: יש לנו 2 מקרים



**מקרה א**

**מקרה ב**

**מסקנה:** אלגוריתם דטרמיניסטי שולח  $n$  ביטים.

**נראה:** אלגוריתם אקראי ששולח  $O(\log n)$  ביטים.

### נסיון ראשון:

אליס: מגרילה  $1 \leq i \leq n$  בהתפלגות אחידה ושולחת  $x_i, i$ . כלומר אליס בוחרת בהתפלגות אחידה כניסה במחרוזת שלה ושולחת את הערך בכניסה זו ואת המיקום

בוב: אם  $x_i = y_i$  מחזיר "שווים"  
אם  $x_i \neq y_i$  מחזיר "שונים"

### ניתוח הפרוטוקול:

- אם  $x = y$  אזי הפרוטוקול תמיד מחזיר "שווים".
- אם  $x \neq y$  אזי קיים לפחות אינדקס אחד עבורו  $x_i \neq y_i$ . בהסתברות לפחות  $\frac{1}{n}$  אליס תגריל אינדקס כזה והפרוטוקול יחזיר תשובה נכונה.

אז אם  $x, y$  שונים בהמון ביטים, אז האלגוריתם יצדק בהסתברות גבוהה. אבל מה קורה אם המחרוזות שונות בביט אחד בלבד? הפרוטוקול צודק בהסתברות קטנה מדי.

מקרה גרוע -  $x, y$  שונים בבדיק ביט אחד.  
הבעיה: ההסת' לטעות עבור  $x \neq y$  גדולה מדי.

שאלה: אולי נחזור על הפרוטוקול הזה מספר פעמים וכך נקבל הסתברות נמוכה יותר לשגיאה?  
תשובה: היינו צריכים לחזור על הפרוטוקול  $\Omega(n)$  פעמים כדי לקבל אפילו הסתברות שגיאה קבועה (למשל  $\frac{1}{4}$ )

דבר טוב ברעיון הזה: התקשורת נמוכה. כדי ליצג אינדקס  $1 \leq i \leq n$  נדרשים רק  $\log n$  ביטים. לכן סה"כ תקשורת בפרוטוקול שניסונו היא  $1 + \log n$ .

## Rabin&Karp 87 פרוטוקול שעובד:

אינטואיציה: הבעיה בניסיון הקודם הייתה ששלחנו ביט שלא מקודד את כל השוני במחרוזת, אלא רק משהו לוקלי

תזכורת: מזה האופרטור  $mod$ ? זה שארית חלוקה. למשל:

$$13 \bmod 7 = 6$$

$$5 \bmod 8 = 5$$

$$8 \bmod 5 = 3$$

הערה:  $x, y$  הן מחרוזות בינאריות באורך  $n$ . נתייחס אליהן כאל מספרים בתחום  $(2^n - 1), \dots, 0, 1, 2, 3, \dots$

### הפרוטוקול:

אליס: מגרילה ראשוני  $2 \leq q \leq n^2$   
מחשבת  $z \leftarrow x \bmod q$  (כאשר כאן אנחנו מתייחסים אל  $x$  כאל מספר המיוצג בייצוג בינארי)  
שולחת  $z, q$  לבוב

בוב: אם  $z \neq y \bmod q$  אזי בוב מחזיר "שונים"  
אחרת בוב מחזיר "שווים"

בדיקת שפיות: מי יותר גדול (באופן טיפוס) מבין  $x, y, q$ ?  
ומה לגבי  $z$ ?

לפני שנוכיח את נכונות הפרוטוקול, ננתח את מספר הביטים שאלים שולחת לבוב.  
מספר הביטים שיידרשו לייצוג  $q$  הוא  $\log(n^2) = 2 \cdot \log(n)$ .  
 $z < q$  ולכן צריך לכל היותר  $2 \cdot \log(n)$  ביטים כדי לייצג את  $z$ .  
 $\Leftarrow$  סה"כ תקשורת:  $4 \cdot \log(n)$

### ניתוח השגיאה של הפרוטוקול:

- אם  $x = y$  אזי לכל  $q$  מתקיים  $x \bmod q = y \bmod q$  ולכן הפרוטוקול תמיד יחזיר "שוים"
- נראה שאם  $x \neq y$  אזי בהסתברות גבוהה הפרוטוקול יחזיר "שונים".

דוגמה:  $x = 53$   $y = 25$   
אם נגריל מספר ראשוני  $q = 7$  אז נקבל

$$\begin{aligned} 53 \bmod 7 &= 4 \\ 25 \bmod 7 &= 4 \end{aligned}$$

ולכן הפרוטוקול יטעה.

לעומת זאת, אם נגריל מספר ראשוני  $q = 5$  אז נקבל

$$\begin{aligned} 53 \bmod 5 &= 3 \\ 25 \bmod 5 &= 0 \end{aligned}$$

והפרוטוקול יענה נכון.

עבור אילו ערכים של  $q$  הפרוטוקול טועה בדוגמה זו? בדיוק עבור ערכי  $q$  המקיימים

$$\begin{aligned} 53 \bmod q &= 25 \bmod q \\ \Downarrow \\ (53 \bmod q) - (25 \bmod q) &= 0 \\ \Downarrow \\ (53 - 25) \bmod q &= 0 \end{aligned}$$

כאשר המעבר האחרון נובע מחוקי ה  $\bmod$  (במידת הצורך, ראו תזכורת בסוף הקובץ). כלומר, הפרוטוקול טועה עבור ערכים של  $q$  כך ש-  $q$  מחלק את  $53 - 25 = 28$ .

**טענה 1:** עבור  $x \neq y$ , הפרוטוקול טועה אם"ם אליו מגרילה  $q$  כך ש-  $q$  מחלק את  $(x - y)$

### הוכחה:

הפרוטוקול טועה אם"ם

$$x \bmod q = y \bmod q$$

כלומר אם"ם

$$(x \bmod q) - (y \bmod q) = 0$$

כלומר אם"ם

$$(x - y) \bmod q = 0$$

כלומר אם"ם  $q$  מחלק את  $(x - y)$ .

**מסקנה:** ההסתברות שהפרוטוקול טועה היא:

$$\Pr[\text{error}] = \frac{|\{q : (x - y) \text{ המחלק את } n^2 \geq q\}|}{|\{q : n^2 \geq q\}|}$$

**טענה 2 (נוכיח עוד רגע):** מספר הראשוניים המחלקים מספר קטן מ-  $2^n$  הוא לכל היותר  $n$

**טענה 3 (ללא הוכחה):** מספר הראשוניים הקטנים שווים ממספר  $t$  הוא  $\theta\left(\frac{t}{\log t}\right)$ . בנוסף, עבור  $t \geq 55$  מתקיים שמספר הראשוניים הקטנים שווים מ-  $t$  הוא לפחות  $\frac{t}{2 \log t}$

**מסקנה:** עבור  $x \neq y$  הפרוטוקול טועה בהסתברות לכל היותר

$$\frac{n}{\left(\frac{n^2}{2 \log(n^2)}\right)} = \frac{4 \log n}{n}$$

ועבור  $n$  מספיק גדול (בערך גדול מ 100) ההסתברות לטעות של הפרוטוקול תהיה לכל היותר  $1/4$

(הערה:  $\frac{1}{4}$  נבחר כקבוע כלשהו. אין כאן משמעות לקבוע)

טענה 3 היא מתורת המספרים ולא נוכיח אותה. מה שהיא אומרת לנו זה שיש יחסית הרבה מספרים ראשוניים.

נוכיח את טענה 2:

יהי  $0 < w < 2^n$  מספר כלשהו, ונסמן ב-  $p_1, p_2, \dots, p_t$  את הראשוניים השונים המחלקים את  $w$ .  
לכן  $p_1 \cdot p_2 \cdot \dots \cdot p_t$  מחלק את  $w$ .

לדוגמה, אם גם 5 וגם 3 מחלקים את 300, אז גם 15 מחלק את 300.

לכן:

$$2^t \leq \underbrace{p_1 \cdot p_2 \cdot \dots \cdot p_t}_{\substack{\text{כי } p_i \geq 2 \\ \text{לכל } i}} \leq w < 2^n$$

לכן מתקיים  $t < n$ , כאשר  $t$  הוא מספר הראשוניים השונים המחלקים את  $w$ .  
מ.ש.ל. (טענה 2)

עד כאן הוכחנו שאם  $x = y$  אז הפרוטוקול תמיד צודק, והוכחנו שאם  $x \neq y$  אז הפרוטוקול טועה בהסתברות לכל היותר  $\frac{1}{4}$ .

**בעייה:**  $\frac{1}{4}$  שגיאה גדולה מדי.

**פתרון:** נריץ את הפרוטוקול  $\ell$  פעמים. כלומר:

אליס: תגריל  $\ell$  ראשוניים  $q_1, \dots, q_\ell$  ותשלח לבוב:

$$q_1, z_1 = x \bmod q_1$$

$$q_2, z_2 = x \bmod q_2$$

...

$$q_\ell, z_\ell = x \bmod q_\ell$$

בוב: אם קיים  $j$  כך שמתקיים  $z_j \neq y \bmod q_j$  אזי עונה "שונים".  
אחרת עונה "שוים".

### ניתוח השגיאה:

אם  $x = y$  אז הפרוטוקול תמיד מחזיר "שוים".

אם  $x \neq y$  אז הפרוטוקול טועה אם כל ה- $q_j$ -ים שבחרנו "מטעים", כלומר  $y \bmod q_j = z_j$ .  
מה ההסתברות שגם  $q_1$  מטעה וגם  $q_2$  מטעה וגם...?

מכיוון שאנו מגרילים את ה- $q_j$ -ים באופן בלתי תלוי אזי ההסתברות שהגיאה היא לכל היותר  $\left(\frac{1}{4}\right)^\ell$ .

דוגמה: אם נרצה ההסתברות שגיאה לכל היותר  $1/2^{100}$  אז ניקח  $\ell = 50$  ונקבל:

$$\left(\frac{1}{4}\right)^{50} = \frac{1}{4^{50}} = \frac{1}{(2^2)^{50}} = \frac{1}{2^{100}}$$

**מה המחיר מבחינת התקשורת של החזרות האלה?**

עבור שגיאה לכל היותר  $\left(\frac{1}{4}\right)^\ell$  נשלח  $4\ell \log n$  ביטים.

כלומר – יש לנו כאן טריידאוף בין התקשורת לשגיאה. אם ניקח  $\ell$  סביר, למשל 50, אז המחיר שאנו משלמים בתקשורת הוא לא יותר מדי גבוה.

מה שראינו כאן היא שיטה שטובה בהרבה מקרים – לוקחים פרוטוקול שנותן הסתברות שגיאה מסויימת ומריצים אותו הרבה פעמים. כאן היה לנו ייתרון – אם  $x = y$  אז הפרוטוקול תמיד מחזיר "שוים". לכן יכולנו להריץ את הפרוטוקול הרבה פעמים והספיק לנו שפעם אחת נקבל "שוים" על מנת לדעת בוודאות ש- $x \neq y$ . במקרה הכללי שבו גם אם  $x = y$  יש הסתברות מסויימת לשגיאה, היינו צריכים לבחור את התשובה שמופיעה הכי הרבה פעמים, ואז חישוב ההסתברות היה נעשה טיפה מסובך יותר.

### ניתוח זמן הריצה של אליס:

- הגרלת ראשוני: נגריל מספר ונבדוק אם הוא ראשוני. אם לא ראשוני אז נחזור על התהליך.
- מה ההסתברות שנגדיל מספר ראשוני?

$$\frac{\text{מספר הראשוניים הקטנים מ } n^2}{n^2} \geq \frac{\left(\frac{n^2}{4 \log n}\right)}{n^2} = \frac{1}{4 \log n}$$

ולכן המספר הממוצע של הגדלות שנבצע לפני שנמצא מספר ראשוני הוא לכל היותר  $4 \log n$ .  
 למה? יש לנו כאן סדרת ניסויים שכל ניסוי מצליח בהסתברות  $p = \frac{1}{4 \log n}$ , ואנחנו שואלים את עצמנו מהי תוחלת מספר הניסויים הדרושים עד להצלחה ראשונה. זה בדיוק משתנה עם התפלגות גיאומטרית, והתחלות שלו היא  $\frac{1}{p} = 4 \log n$ .  
 הסבר נוסף: אם כל ניסוי מצליח בהסתברות  $\frac{1}{4 \log n}$ , אז מהי ההסתברות שנבצע  $4 \log n$  ניסויים וכולם ייכשלו? לכל היותר

$$\left(1 - \frac{1}{4 \log n}\right)^{4 \log n} \leq \left(e^{-\frac{1}{4 \log n}}\right)^{4 \log n} = e^{-1} = \frac{1}{e}$$

• איך נבדוק האם המספר שהוגדל הוא ראשוני?

אפשר לעבור על כל המספרים בין 2 ל-  $\sqrt{q}$  ולבדוק אבל זה לא יעיל.

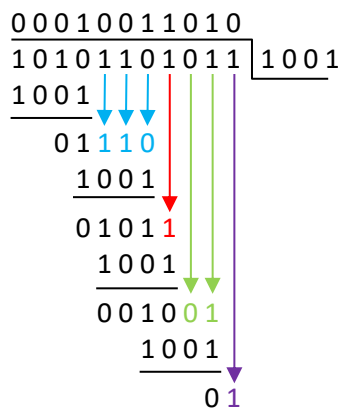
**טענה:** קיים אלגוריתם אקראי שרץ בזמן  $O(\log^3 q)$  ובודק האם  $q$  הוא ראשוני

← סה"כ הגרלת מספר ראשוני  $q$  קטן שווה מ-  $n^2$  מתבצעת בזמן  $(\log^4 n)$  בממוצע.

• חישוב  $x \bmod q$  (וגם  $y \bmod q$ ) מתבצע ע"י חילוק ארוך. נזכור ש-  $x$  הוא מספר בן  $n$  ביטים ו-  $q$  הוא מספר בן  $2 \log n$  ביטים. לכן תהליך החילוק הארוך מתבצע בזמן  $O(n)$

• סה"כ זמן ריצה  $O(n)$

דוגמה לחילוק ארוך:



**הערה:** תהליך החילוק הארוך כמו שהוא תואר פה אורך זמן  $O(n \cdot \log n)$  כדי לחלק את  $x$  ב-  $q$ .

**אבל:** למרות ש-  $n$  הוא מספר גדול מאוד, עדיין אנחנו מניחים ש-  $\log n$  ביטים נכנסים לתא זכרון בודד של מחשב.

למה זאת הנחה סבירה?  
 כי גם אם הקלט שלי הוא באורך  $n = 2^{64}$  ביטים, שזה בערך 2,000,000,000 גיגה בית,  
 אז עדיין  $\log n = 64$  ביטים נכנסים בתא זיכרון אחד של המחשב.

**הנחה:** אם גודל הקלט הוא  $n$  ביטים, אזי פעולות אריטמטיות על מספרים בני  $\log n$  ביטים מתבצעות בזמן קבוע.

למעשה, כבר הייתה לנו את ההנחה הסומייה הזאת לאורך כל הקורס ופשוט לא שמנו לב אליה.  
 למשל, כמה זמן לוקח להעתיק גרף?  $O(|E| + |V|)$ , נכון?  
 אבל אנחנו מייצגים גרפים ע"י רשימת שכנויות, אז לכל קודקוד אני צריך פויינטר. כלומר אני צריך  $|V|$  פויינטרים.  
 אז כדי לייצג כל פויינטר אני צריך  $\log|V|$  ביטים. מה קורה אם  $|V|$  כל כך גדול ש-  $\log|V|$  לא נכנס בתא זיכרון?  
 אז יש לי  $|V|$  קודקודים ולכל קוד' אני צריך פויינטר שדורש אסימפטוטית  $\log|V|$  תאי זיכרון...

תזכורת: איך רואים שמתקיים  $x \bmod q = y \bmod q$  אם ורק אם  $(x - y) \bmod q = 0$ ?  
 תשובה: נסמן

$$x = a_1 \cdot q + b_1$$

$$y = a_2 \cdot q + b_2$$

עבור שלמים  $a_1, a_2, b_1, b_2$  כך ש-  $0 \leq b_1, b_2 < q$ .  
 לפי הגדרה,

$$x \bmod q = b_1$$

$$y \bmod q = b_2$$

נעת, אם מתקיים  $x \bmod q = y \bmod q$ , כלומר  $b_1 = b_2$ , אזי מתקיים

$$(x - y) \bmod q = (a_1 \cdot q + b_1 - a_2 \cdot q - b_2) \bmod q = (b_1 - b_2) \bmod q$$

ובכיוון השני, אם  $(x - y) \bmod q = 0$  אזי  $(x - y)$  הוא כפולה של  $q$ . כלומר קיים  $k$  שלם כך ש-

$$(1) \quad x - y = (a_1 - a_2) \cdot q + \underbrace{(b_1 - b_2)}_{\text{מה זה?}} = k \cdot q$$

נזכור ש-  $0 \leq b_1, b_2 < q$  ולכן  $-q < (b_1 - b_2) < q$  ולכן היחידה כך שאגף שמאל במשוואה (1) יהיה כפולה שלמה של  $q$  היא אם  $(b_1 - b_2) = 0$ , כלומר  $x \bmod q = y \bmod q$ .