אלגוריתמים אקראיים

חזרה קצרה | תרגילים

חזרה קצרה – סוגי אלגוריתמים אקראיים

אלגוריתם לאס וגאס

אלגוריתם אקראי **המחזיר תמיד תוצאה נכונה** (או מודיע על כישלון). במילים אחרות, אלגוריתם לאס וגאס אינו מהמר על נכונות התוצאה, הוא **מהמר רק על המשאבים** המשמשים לצורך החישוב (למשל, זמן החישוב).

דוגמה: מיון מהיר אקראי, כאשר איבר הציר נבחר באופן אקראי.

התוצאה תמיד ממוינת, אך זמן הריצה נעשה מהיר יותר בהסתברות מסוימת.

אלגוריתם מונטה קרלו

אלגוריתם אקראי **שהתוצאה שלו עלולה להיות שגויה** בהסתברות מסוימת, אך **זמן הריצה שלו חסום.**

תרגיל 1

נתון מקור של ביטים אקראיים לא מוטים (כלומר $p(1)=p(0) = \frac{1}{2}$).

כיצד ניתן להשתמש בהם כדי ליצור דגימה אחידה מהקבוצה $S = \{0,1,2,....,n-1\}$?

מהי תוחלת מספר הביטים האקראיים הנדרשים לאלגוריתם?

תרגיל 1

נתון מקור של ביטים אקראיים לא מוטים (כלומר $p(1)=p(0) = \frac{1}{2}$).

כיצד ניתן להשתמש בהם כדי ליצור דגימה אחידה מהקבוצה $S = \{0,1,2,....,n-1\}$?

מהי תוחלת מספר הביטים האקראיים הנדרשים לאלגוריתם?

לדוגמה:

$n = 10$
$\log N = 4$
$N = 16$

תרגיל 1

נתון מקור של ביטים אקראיים לא מוטים (כלומר $p(1)=p(0) = \frac{1}{2}$).

כיצד ניתן להשתמש בהם כדי ליצור דגימה אחידה מהקבוצה $S = \{0,1,2,....,n-1\}$?

מהי תוחלת מספר הביטים האקראיים הנדרשים לאלגוריתם?

לדוגמה:

n = 10
logN = 4
N = 16

פתרון

עבור n כלשהו, נגריל $\lceil \log n \rceil$ ביטים.

נסמן: $N = 2^{\lceil \log n \rceil}$, כלומר, נגריל logN ביטים.

מתוך הדגימה שנגריל, נוכל לקבל כל מספר מהקבוצה $S = \{0,1,2,\ldots,n-1\}$, אך בגלל שעיגלנו כלפי מעלה,

נוכל לקבל גם מספרים שאינם בתחום הרצוי.

logN ביטים נותנים מספרים בתחום 0,...,N-1 בהסתברות $\frac{1}{N}$ לכל מספר, כולל למספרים שאינם בתחום.

אנו צריכים לייצר מספרים בתחום הנכון בסיכויי שווה, כלומר בסיכוי של $\frac{1}{n}$ כאשר ידוע ש $\frac{1}{N} \leq \frac{1}{n}$.

לכן, אם קיבלנו מספר שאינו בתחום, נגריל logN ביטים חדשים. נמשיך בתהליך עד לקבלת מספר קטן מ-n. כעת, אם ב-logN הביטים

האחרונים התקבל מספר בתחום הרצוי, הסיכוי שזה היה מספר ספציפי k הוא $\frac{1}{n}$ (הסיכוי לא תלוי בנסיונות קודמים)

תרגיל 1

תזכורת - הסתברות

<u>הסתברות מותנית</u>: ההסתברות ל-A בהנחה/בתנאי ש-B

$$\frac{P(A\cap B)}{P(B)}$$

דוגמה: בהינתן קובייה הוגנת – בהנחה שהוגרל מספר זוגי – מה ההסתברות שיהיה גדול מ4?

$$\frac{P(\text{מספר זוגי וגדול מ4})}{P(\text{מספר זוגי})} = \frac{1/6}{3/6} = \frac{1}{3}$$

<u>ניסוי ברנולי</u>: ניסוי בו יש שתי תוצאות אפשריות – הצלחה או כשלון.

<u>משתנה מקרי גאומטרי</u>: כאשר מבצעים סדרה של ניסויי ברנולי עד לקבלת ההצלחה הראשונה, וסופרים כמה ניסויים ביצענו, אז מספר הניסויים שביצענו נקרא <u>משתנה מקרי גאומטרי.</u>

תוחלת משתנה מקרי גאומטרי היא $\frac{1}{P}$, כאשר P היא ההסתברות להצלחה.

תרגיל 1

נתון מקור של ביטים אקראיים לא מוטים (כלומר $p(1)=p(0) = \frac{1}{2}$).

כיצד ניתן להשתמש בהם כדי ליצור דגימה אחידה מהקבוצה $S = \{0,1,2,\ldots,n-1\}$?

מהי תוחלת מספר הביטים האקראיים הנדרשים לאלגוריתם?

ניתוח הסתברותי

הסיכוי לכל מספר:

ההסתברות לבחירת מספר x ספציפי בתחום 0, 1, …., n-1 ב-logN הביטים האחרונים (בהנחה שב-logN הביטים האחרונים יצא מספר בתחום זה)

$$P(x = k \mid x < n) = \frac{P(x=k)}{P(x<n)} = \frac{\frac{1}{N}}{\frac{n}{N}} = \frac{1}{n}$$

הדגימה אחידה

תוחלת מספר הביטים

<u>תוחלת מספר הנסיונות</u>: $\frac{1}{\frac{n}{N}} = \frac{N}{n}$, מספר הביטים לכל נסיון logN.

$\frac{N}{n} < 2$, אחרת ניקח חזקה קטנה יותר של 2.

<u>תוחלת מספר הביטים</u>: $\frac{N}{n}\log N = O(\log n)$.

תרגיל 2

נתונות 3 מטריצות $n \times n$ A,B,C. כתבו אלגוריתם אקראי שבודק האם $A \cdot B = C$.

הפתרון הנאיבי

נכפול את המטריצות ונבדוק האם התוצאה זהה ל-C.

עלות: $O(n^3)$

תרגיל 2

נתונות 3 מטריצות $n \times n$ A,B,C. כתבו אלגוריתם אקראי שבודק האם $A \cdot B = C$.

הפתרון הנאיבי

נכפול את המטריצות ונבדוק האם התוצאה זהה ל-C.

עלות: $O(n^3)$

פתרון אקראי

<u>הרעיון:</u>

נגריל וקטור אקראי v ונחשב את $A \cdot (B \cdot v)$ ו-$C \cdot v$ ונשווה ביניהם.

נעדיף להגריל וקטור בוליאני, משום שאנו מניחים שהגרלת ביט אחד עולה $O(1)$.

תרגיל 2

נתונות 3 מטריצות $n \times n$ A,B, C. כתבו אלגוריתם אקראי שבודק האם $A \cdot B = C$.

פתרון אקראי

<u>הרעיון:</u>

נגריל וקטור אקראי v ונחשב את $A \cdot (B \cdot v)$ ו-$C \cdot v$ ונשווה ביניהם.

נעדיף להגריל וקטור בוליאני, משום שאנו מניחים שהגרלת ביט אחד עולה $O(1)$.

<u>האלגוריתם:</u>

<u>check_multiple_matrix</u>(A,B,C, n):

1. v ⟵get_random_binary_vector(n)
2. Bv ⟵ multiple_matrix(B,v)
3. ABv ⟵ multiple_matrix(A,Bv)
4. Cv ⟵ multiple_matrix(C,v)
5. **if** is_equal(Cv, ABv):
6. **then** return ”Equal”
7. **else** return ”Not-Equal”

תרגיל 2

נתונות 3 מטריצות $n \times n$ A,B, C. כתבו אלגוריתם אקראי שבודק האם $A \cdot B = C$.

פתרון אקראי

<u>הרעיון:</u>

נגריל וקטור אקראי v ונחשב את $A \cdot (B \cdot v)$ ו-$C \cdot v$ ונשווה ביניהם.

נעדיף להגריל וקטור בוליאני, משום שאנו מניחים שהגרלת ביט אחד עולה $O(1)$.

<u>האלגוריתם:</u>

<u>check_multiple_matrix</u>(A,B,C, n):

$O(n)$	1. v ⟵get_random_binary_vector(n)
$O(n^2)$	2. Bv ⟵ multiple_matrix(B,v)
$O(n^2)$	3. ABv ⟵ multiple_matrix(A,Bv)
$O(n^2)$	4. Cv ⟵ multiple_matrix(C,v)
$O(n)$	5. **if** is_equal(Cv, ABv):
$O(1)$	6. **then** return ”Equal”
	7. **else** return ”Not-Equal”

עלות: $O(n^2)$

תרגיל 2

נתונות 3 מטריצות $n \times n$ A,B, C. כתבו אלגוריתם אקראי שבודק האם $A \cdot B = C$.

פתרון אקראי

<u>הרעיון:</u>

נגריל וקטור אקראי v ונחשב את $A \cdot (B \cdot v)$ ו-$C \cdot v$ ונשווה ביניהם.

נעדיף להגריל וקטור בוליאני, משום שאנו מניחים שהגרלת ביט אחד עולה $O(1)$.

האם האלגוריתם תמיד יחזיר תשובה נכונה?

<u>check_multiple_matrix</u>(A,B,C, n):

```
v ⟵get_random_binary_vector(n)
Bv ⟵ multiple_matrix(B,v)
ABv ⟵ multiple_matrix(A,Bv)
Cv ⟵ multiple_matrix(C,v)
 if is_equal(Cv, ABv):
    then return ”Equal”
    else  return  ”Not-Equal”
```

עלות: $O(n^2)$

תרגיל 2

נתונות 3 מטריצות $n \times n$ A,B, C. כתבו אלגוריתם אקראי שבודק האם $A \cdot B = C$.

פתרון אקראי

<u>הרעיון:</u>

נגריל וקטור אקראי v ונחשב את $A \cdot (B \cdot v)$ ו-$C \cdot v$ ונשווה ביניהם.

נעדיף להגריל וקטור בוליאני, משום שאנו מניחים שהגרלת ביט אחד עולה O(1).

check_multiple_matrix(A,B,C, n):

1. v ⟵get_random_binary_vector(n)
2. Bv ⟵ multiple_matrix(B,v)
3. ABv ⟵ multiple_matrix(A,Bv)
4. Cv ⟵ multiple_matrix(C,v)
5. **if** is_equal(Cv, ABv):
6. **then** return ”Equal”
7. **else** return ”Not-Equal”

עלות: $O(n^2)$

האם האלגוריתם תמיד יחזיר תשובה נכונה?

<u>נסתכל על המצבים האפשריים:</u> לשם הנוחות, נסמן: $D = AB-C$, במידה והשוויון מתקיים, $D=0$ וגם כמובן $D(v) = 0$

האלגוריתם מחזיר "שווים"

$ABv = Cv$ - האלגוריתם יחזיר ”Equal”

1. $AB=C, D=0$ ✅
2. $AB \neq C, D \neq 0$ – נרצה להראות שההסתברות למקרה זה קטנה או שווה ל-$\frac{1}{2}$ ❌

האלגוריתם מחזיר "לא שווים"

$ABv \neq Cv$ - האלגוריתם יחזיר ”Not-Equal” ✅

תרגיל 2

נתונות 3 מטריצות $n \times n$ A,B, C. כתבו אלגוריתם אקראי שבודק האם $A \cdot B = C$.

פתרון אקראי

<u>הרעיון:</u>

נגריל וקטור אקראי v ונחשב את $A \cdot (B \cdot v)$ ו-$C \cdot v$ ונשווה ביניהם.

נעדיף להגריל וקטור בוליאני, משום שאנו מניחים שהגרלת ביט אחד עולה O(1).

check_multiple_matrix(A,B,C, n):

1. v ⟵get_random_binary_vector(n)
2. Bv ⟵ multiple_matrix(B,v)
3. ABv ⟵ multiple_matrix(A,Bv)
4. Cv ⟵ multiple_matrix(C,v)
5. **if** is_equal(Cv, ABv):
6. **then** return "Equal"
7. **else** return "Not-Equal"

עלות: $O(n^2)$

האם האלגוריתם תמיד יחזיר תשובה נכונה?

<u>נסתכל על המצבים האפשריים:</u> לשם הנוחות, נסמן: D = AB-C, במידה והשוויון מתקיים, D=0 וגם כמובן D(v) = 0

דוגמה: D = AB-C v – וקטור שהוגרל אקראית

תשובת האלגוריתם

D v

לא שווים $\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$

שווים $\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$

האלגוריתם מחזיר "שווים"

$ABv = Cv$ - האלגוריתם יחזיר "Equal"

1. AB=C, D=0
2. AB≠C, D≠0 – נרצה להראות שההסתברות למקרה זה קטנה או שווה ל-$\frac{1}{2}$

האלגוריתם מחזיר "לא שווים"

$ABv \neq Cv$ - האלגוריתם יחזיר "Not-Equal"

תרגיל 2

נתונות 3 מטריצות $n\times n$ A,B, C. כתבו אלגוריתם אקראי שבודק האם $A\cdot B = C$.

<u>נסתכל על המצבים האפשריים:</u> לשם הנוחות, נסמן: $D = AB-C$, במידה והשוויון מתקיים, $D=0$ וגם כמובן $D(v) = 0$

דוגמה: $D = AB-C$ v – וקטור שהוגרל אקראית

תשובת האלגוריתם v D

לא שווים $\begin{pmatrix}1 & -1\\ -1 & 1\end{pmatrix}\cdot\begin{pmatrix}1\\0\end{pmatrix}=\begin{pmatrix}1\\-1\end{pmatrix}$

שווים $\begin{pmatrix}1 & -1\\ -1 & 1\end{pmatrix}\cdot\begin{pmatrix}1\\1\end{pmatrix}=\begin{pmatrix}0\\0\end{pmatrix}$

האלגוריתם מחזיר "שווים"

$ABv = Cv$ - האלגוריתם יחזיר "Equal"

1. $AB=C, D=0$
2. $AB\neq C, D\neq 0$ – נרצה להראות שההסתברות למקרה זה קטנה או שווה ל-$\frac{1}{2}$

הגדרה: עבור $D\neq 0$

<u>"וקטור שקרן"</u> - וקטור בוליאני v באורך n המקיים $D\cdot v = 0$.

המשמעות היא שניסינו להגריל וקטור שבעזרתו נוכל לזהות ש-$AB\neq C$ אך יצא וקטור שלא מראה זאת.

הבחנה:

אם $AB\neq C$ אז ההסתברות לטעות שקולה להסתברות להגריל "וקטור שקרן".

היות ו- $D\neq 0$, קיים איבר ב-D ששונה מ-0. נסמן איבר זה ב$d_{i,j}$ וכן נסמן בD^i את השורה ה-i במטריצה D.

נבחר וקטור $v=(v_1,v_2,\dots,v_n)$ שבו כל הכניסות **פרט ל-v_j** נקבעו מראש.

תרגיל 2

נתונות 3 מטריצות $n\times n$ A,B,C. כתבו אלגוריתם אקראי שבודק האם $A\cdot B = C$.

<u>נסתכל על המצבים האפשריים:</u> לשם הנוחות, נסמן: $D = AB-C$, במידה והשוויון מתקיים, $D=0$ וגם כמובן $D(v) = 0$

האלגוריתם מחזיר "שווים"

$ABv = Cv$ - האלגוריתם יחזיר "Equal"

1. $AB=C,\ D=0$
2. $AB\neq C,\ D\neq 0$ – נרצה להראות שההסתברות למקרה זה קטנה או שווה ל-$\frac{1}{2}$

$$D = \begin{array}{c c} & j \\ i & \begin{pmatrix} & & \\ & 1 & \\ & & \end{pmatrix} \end{array} \begin{array}{c} v \\ \begin{pmatrix} \\ v_j \\ \\ \end{pmatrix} \end{array}$$

הגדרה: עבור $D\neq 0$

<u>"וקטור שקרן"</u> - וקטור בוליאני v באורך n המקיים $D\cdot v = 0$.
המשמעות היא שניסינו להגריל וקטור שבעזרתו נוכל לזהות ש-$AB\neq C$ אך יצא וקטור שלא מראה זאת.

הבחנה:

אם $AB\neq C$ אז ההסתברות לטעות שקולה להסתברות להגריל "וקטור שקרן".
היות ו- $D\neq 0$, קיים איבר ב-D ששונה מ-0. נסמן איבר זה ב$d_{i,j}$ וכן נסמן בD^i את השורה ה-i במטריצה D.
נבחר וקטור $v=(v_1,v_2,\dots,v_n)$ שבו כל הכניסות **פרט ל-v_j** נקבעו אקראית מראש.

הרכיב v_j תלוי בבחירה של הרכיבים האחרים בוקטור v. דרגת החופש ירדה ל$n-1$. לאחר שבחרנו את שאר הרכיבים v_i $i\neq j$.
בעצם את ערך v_j נהיה חייבים לקבוע לפי הערכים שנקבעו קודם (אם ייבחר אקראית, לא בטוח שנקבל "וקטור שקרן")

תרגיל 2

תזכורת - הסתברות

בהינתן קובייה הוגנת – מה ההסתברות להוציא בהטלה אחת את הספרה 6?

בהינתן קובייה הוגנת – מה ההסתברות להוציא בשתי הטלות רצופות את הספרה 6?

תרגיל 2

תזכורת - הסתברות

בהינתן קובייה הוגנת – מה ההסתברות להוציא בהטלה אחת את הספרה 6?

$$\frac{1}{6} = \frac{\text{האפשרויות שבהן אנחנו מעוניינים}}{\text{סך כל האפשרויות הקיימות}}$$

בהינתן קובייה הוגנת – מה ההסתברות להוציא בשתי הטלות רצופות את הספרה 6?

הסיכוי להוציא 6 בהטלה <u>הראשונה</u> **וגם** להוציא 6 בהטלה <u>השנייה</u>

$$\frac{1}{6} \times \frac{1}{6} = \frac{1}{36}$$

תרגיל 2

$$D = \begin{array}{c c} & j \\ i & \begin{pmatrix} & & \\ & 1 & \\ & & \end{pmatrix} \end{array} \begin{array}{c} v \\ \begin{pmatrix} \\ v_j \\ \\ \end{pmatrix} \end{array}$$

נתונות 3 מטריצות $n \times n$ A,B,C. כתבו אלגוריתם אקראי שבודק האם $A \cdot B = C$.

הגדרה: עבור $D \neq 0$

"וקטור שקרן" - וקטור בוליאני v באורך n המקיים $D \cdot v = 0$.

המשמעות היא שניסינו להגריל וקטור שבעזרתו נוכל לזהות ש-$AB \neq C$ אך יצא וקטור שלא מראה זאת.

הבחנה:

אם $AB \neq C$ אז ההסתברות לטעות שקולה להסתברות להגריל "וקטור שקרן".

היות ו- $D \neq 0$, קיים איבר ב-D ששונה מ-0. נסמן איבר זה ב$d_{i,j}$ וכן נסמן בD^i את השורה ה-i במטריצה D.

נבחר וקטור $v=(v_1,v_2,\dots,v_n)$ שבו כל הכניסות **פרט ל-v_j** נקבעו אקראית מראש.

הרכיב v_j תלוי בבחירה של הרכיבים האחרים בוקטור v. דרגת החופש ירדה ל$n-1$. לאחר שבחרנו את שאר הרכיבים v_i $i \neq j$.

בעצם את ערך v_j נהיה חייבים לקבוע לפי הערכים שנקבעו קודם (אם ייבחר אקראית, לא בטוח שנקבל "וקטור שקרן")

ההסתברות P לכישלון (לטעות) היא מספר הווקטורים השקרנים **מתוך** מספר כלל האפשרויות לבחור וקטור באורך המתאים

$$P(\text{כישלון}) = \frac{\text{מספר ה"וקטורים השקרנים"}}{\text{מספר כלל הוקטורים שניתן להגריל}} =$$

תרגיל 2

$$D = \begin{array}{c c} & j \\ i & \begin{pmatrix} & & \\ & 1 & \\ & & \end{pmatrix} \end{array} \begin{array}{c} v \\ \begin{pmatrix} \\ v_j \\ \\ \end{pmatrix} \end{array}$$

נתונות 3 מטריצות $n \times n$ A,B,C. כתבו אלגוריתם אקראי שבודק האם $A \cdot B = C$.

הגדרה: עבור $D \neq 0$

"וקטור שקרן" - וקטור בוליאני v באורך n המקיים $D \cdot v = 0$.

המשמעות היא שניסינו להגריל וקטור שבעזרתו נוכל לזהות ש-$AB \neq C$ אך יצא וקטור שלא מראה זאת.

הבחנה:

אם $AB \neq C$ אז ההסתברות לטעות שקולה להסתברות להגריל "וקטור שקרן".

היות ו- $D \neq 0$, קיים איבר ב-D ששונה מ-0. נסמן איבר זה ב$d_{i,j}$ וכן נסמן בD^i את השורה ה-i במטריצה D.

נבחר וקטור $v=(v_1,v_2,\dots,v_n)$ שבו כל הכניסות **פרט ל-v_j** נקבעו אקראית מראש.

הרכיב v_j תלוי בבחירה של הרכיבים האחרים בוקטור v. דרגת החופש ירדה ל$n-1$. לאחר שבחרנו את שאר הרכיבים v_i $i \neq j$.

בעצם את ערך v_j נהיה חייבים לקבוע לפי הערכים שנקבעו קודם (אם ייבחר אקראית, לא בטוח שנקבל "וקטור שקרן")

ההסתברות P לכישלון (לטעות) היא מספר הווקטורים השקרנים **מתוך** מספר כלל האפשרויות לבחור וקטור באורך המתאים

$$P(\text{כישלון}) = \frac{\text{מספר ה"וקטורים השקרנים"}}{\text{מספר כלל הוקטורים שניתן להגריל}} = \frac{2^{n-1}}{2^n} = \frac{1}{2}$$

תרגיל 2

נתונות 3 מטריצות $n \times n$ A,B, C. כתבו אלגוריתם אקראי שבודק האם $A \cdot B = C$.

$$D = \begin{array}{c c} & j \\ i & \begin{pmatrix} & & \\ & 1 & \\ & & \end{pmatrix} \end{array} \begin{array}{c} v \\ \begin{pmatrix} \\ v_j \\ \\ \end{pmatrix} \end{array}$$

הגדרה: עבור $D \neq 0$

"וקטור שקרן" - וקטור בוליאני v באורך n המקיים $D \cdot v = 0$.

המשמעות היא שניסינו להגריל וקטור שבעזרתו נוכל לזהות ש-$AB \neq C$ אך יצא וקטור שלא מראה זאת.

ההסתברות P לכישלון (לטעות) היא מספר הווקטורים השקרנים **מתוך** מספר כלל האפשרויות לבחור וקטור באורך המתאים

$$P(\text{כישלון}) = \frac{\text{מספר ה"וקטורים השקרנים"}}{\text{מספר כלל הוקטורים שניתן להגריל}} = \frac{2^{n-1}}{2^n} = \frac{1}{2}$$

האם ניתן לשפר את ההסתברות לכישלון?

תרגיל 2

$$D = \begin{array}{c c} & j \\ i & \begin{pmatrix} & & \\ & 1 & \\ & & \end{pmatrix} \end{array} \begin{array}{c} v \\ \begin{pmatrix} \\ v_j \\ \\ \end{pmatrix} \end{array}$$

נתונות 3 מטריצות $n \times n$ A,B, C. כתבו אלגוריתם אקראי שבודק האם $A \cdot B = C$.

הגדרה: עבור $D \neq 0$

"וקטור שקרן" - וקטור בוליאני v באורך n המקיים $D \cdot v = 0$.

המשמעות היא שניסינו להגריל וקטור שבעזרתו נוכל לזהות ש-$AB \neq C$ אך יצא וקטור שלא מראה זאת.

ההסתברות P לכישלון (לטעות) היא מספר הווקטורים השקרנים **מתוך** מספר כלל האפשרויות לבחור וקטור באורך המתאים

$$P(\text{כישלון}) = \frac{\text{מספר ה"וקטורים השקרנים"}}{\text{מספר כלל הוקטורים שניתן להגריל}} = \frac{2^{n-1}}{2^n} = \frac{1}{2}$$

האם ניתן לשפר את ההסתברות לכישלון?

נריץ את האלגוריתם X פעמים. נקבל:

1. AB=C – בכל ההרצות נקבל "שווים"
2. AB ≠C – ההסתברות שכל ההרצות יטעו (ויענו "שווים") היא: $\frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} \ldots\ldots \cdot \frac{1}{2} = \frac{1}{2^x}$

זמן הריצה $O(x \cdot n^2)$ – יעיל רק עבור x שקטן אסימפטוטית מ-n

תרגיל 2

נתונות 3 מטריצות $n \times n$ A,B, C. כתבו אלגוריתם אקראי שבודק האם $A \cdot B = C$.

האם האלגוריתם הזה הוא אלגוריתם לאס וגאס או מונטה קרלו?

אלגוריתם לאס וגאס

אלגוריתם אקראי **המחזיר תמיד תוצאה נכונה**,הוא **מהמר רק על המשאבים** המשמשים לצורך החישוב (למשל, זמן החישוב).

אלגוריתם מונטה קרלו

אלגוריתם אקראי **שהתוצאה שלו עלולה להיות שגויה** בהסתברות מסוימת, אך **זמן הריצה שלו חסום.**

check_multiple_matrix(A,B,C, n):

```
1.  v ⟵get_random_binary_vector(n)
2.  Bv ⟵ multiple_matrix(B,v)
3.  ABv ⟵ multiple_matrix(A,Bv)
4.  Cv ⟵ multiple_matrix(C,v)
5.   if is_equal(Cv, ABv):
6.      then return ”Equal”
7.      else  return  ”Not-Equal”
```

עלות: $O(x \cdot n^2)$

תרגיל 2

נתונות 3 מטריצות $n \times n$ A,B, C. כתבו אלגוריתם אקראי שבודק האם $A \cdot B = C$.

האם האלגוריתם הזה הוא אלגוריתם לאס וגאס או מונטה קרלו?

מונטה קרלו!

אלגוריתם לאס וגאס

אלגוריתם אקראי **המחזיר תמיד תוצאה נכונה**,הוא **מהמר רק על המשאבים** המשמשים לצורך החישוב (למשל, זמן החישוב).

אלגוריתם מונטה קרלו

אלגוריתם אקראי **שהתוצאה שלו עלולה להיות שגויה** בהסתברות מסוימת, אך **זמן הריצה שלו חסום.**

```
check_multiple_matrix(A,B,C, n):
1.  v ⟵get_random_binary_vector(n)
2.  Bv ⟵ multiple_matrix(B,v)
3.  ABv ⟵ multiple_matrix(A,Bv)
4.  Cv ⟵ multiple_matrix(C,v)
5.   if is_equal(Cv, ABv):
6.      then return "Equal"
7.      else  return  "Not-Equal"
```

עלות: $O(x \cdot n^2)$

האלגוריתם ירוץ X פעמים (לפי X קבוע שנקבע מראש) – זמן הריצה יהיה חסום $O(x \cdot n^2)$

יש סיכוי (קטן) שהתוצאה לא תהיה נכונה.

תרגיל 3

נתון מטבע בעל הסתברות P ל"פלי", כאשר P לא ידועה.

הראו כיצד ליצור מטבע לא מוטה שבו:

$$P(\text{פלי}) = P(\text{עץ}) = \frac{1}{2}$$

תנו סכימה שבה תוחלת מספר ההטלות של המטבע הנתון כדי לקבל הטלה אחת של המטבע הלא מוטה הינה לא יותר מאשר $\frac{1}{P(1-P)}$

תרגיל 3

נתון מטבע בעל הסתברות P ל"פלי", כאשר P לא ידועה.

הראו כיצד ליצור מטבע לא מוטה שבו:

$$P(\text{פלי}) = P(\text{עץ}) = \frac{1}{2}$$

תנו סכימה שבה תוחלת מספר ההטלות של המטבע הנתון כדי לקבל הטלה אחת של המטבע הלא מוטה הינה לא יותר מאשר $\frac{1}{P(1-P)}$

רמז: התבוננו בשתי הטלות רצופות של המטבע הנתון.

תרגיל 3

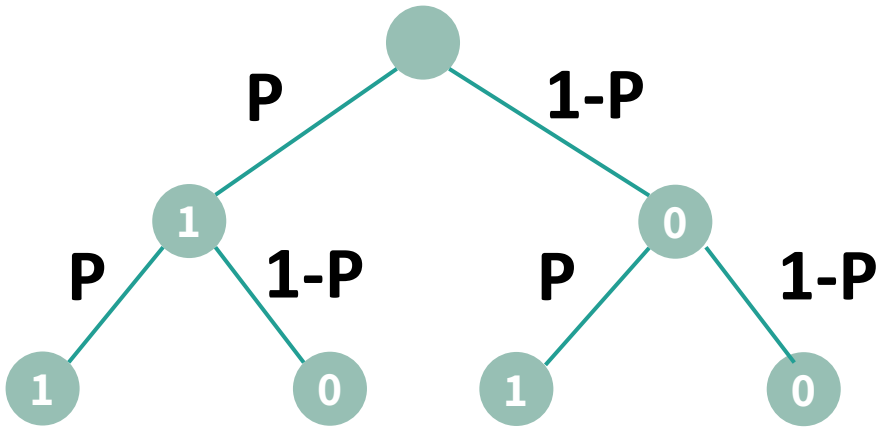
נתון מטבע בעל הסתברות P ל"פלי", כאשר P לא ידועה.

הראו כיצד ליצור מטבע לא מוטה שבו:

$$P(\text{פלי}) = P(\text{עץ}) = \frac{1}{2}$$

תנו סכימה שבה תוחלת מספר ההטלות של המטבע הנתון כדי לקבל הטלה אחת של המטבע הלא מוטה הינה לא יותר מאשר $\frac{1}{P(1-P)}$

רמז: התבוננו בשתי הטלות רצופות של המטבע הנתון.




```
create_fair_coin(coin)
prev_flip ⟵ flip(coin)
current_flip ⟵ flip(coin)
while prev_flip = current_flip:
    do prev_flip ⟵ flip(coin)
        current_flip ⟵ flip(coin)
return current_flip
```

נגדיר "פלי" = 1

תרגיל 3

נתון מטבע בעל הסתברות P ל"פלי", כאשר P לא ידועה.

הראו כיצד ליצור מטבע לא מוטה שבו:

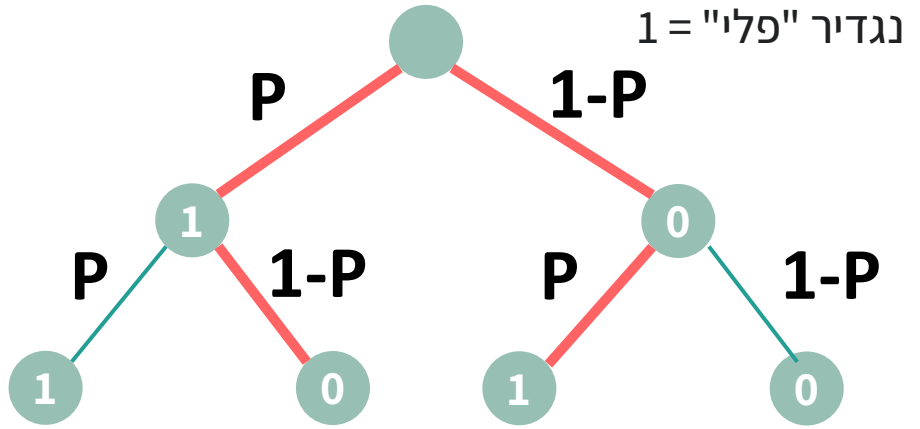
$$P(\text{פלי}) = P(\text{עץ}) = \frac{1}{2}$$

תנו סכימה שבה תוחלת מספר ההטלות של המטבע הנתון כדי לקבל הטלה אחת של המטבע הלא מוטה הינה לא יותר מאשר $\frac{1}{P(1-P)}$

רמז: התבוננו בשתי הטלות רצופות של המטבע הנתון.

create_fair_coin(coin)

1. prev_flip ⟵ flip(coin)
2. current_flip ⟵ flip(coin)
3. **while** prev_flip = current_flip:
4. **do** prev_flip ⟵ flip(coin)
5. current_flip ⟵ flip(coin)
6. **return** current_flip



$P(\text{שתי תוצאות שונות}) = P(1-P) + (1-P)P = 2P(1-P)$

תרגיל 3

נתון מטבע בעל הסתברות P ל"פלי", כאשר P לא ידועה.

הראו כיצד ליצור מטבע לא מוטה שבו:

$$P(\text{פלי}) = P(\text{עץ}) = \frac{1}{2}$$

תנו סכימה שבה תוחלת מספר ההטלות של המטבע הנתון כדי לקבל הטלה אחת של המטבע הלא מוטה הינה לא יותר מאשר $\frac{1}{P(1-P)}$

רמז: התבוננו בשתי הטלות רצופות של המטבע הנתון.

ניתוח הסתברותי:

```
create_fair_coin(coin)
prev_flip ⟵ flip(coin)
current_flip ⟵ flip(coin)
while prev_flip = current_flip:
    do prev_flip ⟵ flip(coin)
       current_flip ⟵ flip(coin)
return current_flip
```

תרגיל 3

תזכורת - הסתברות

<u>הסתברות מותנית</u>: ההסתברות ל-A בהנחה/בתנאי ש-B

$$\frac{P(A\cap B)}{P(B)}$$

דוגמה: בהינתן קובייה הוגנת – בהנחה שהוגרל מספר זוגי – מה ההסתברות שיהיה גדול מ4?

$$\frac{P(\text{מספר זוגי וגדול מ4})}{P(\text{מספר זוגי})} = \frac{1/6}{3/6} = \frac{1}{3}$$

<u>ניסוי ברנולי</u>: ניסוי בו יש שתי תוצאות אפשריות – הצלחה או כשלון.

<u>משתנה מקרי גאומטרי</u>: כאשר מבצעים סדרה של ניסויי ברנולי עד לקבלת ההצלחה הראשונה, וסופרים כמה ניסויים ביצענו, אז מספר הניסויים שביצענו נקרא <u>משתנה מקרי גאומטרי</u>.

תוחלת משתנה מקרי גאומטרי היא $\frac{1}{P}$, כאשר P היא ההסתברות להצלחה.

תרגיל 3

```
create_fair_coin(coin)
prev_flip ⟵ flip(coin)
current_flip ⟵ flip(coin)
while prev_flip = current_flip:
    do prev_flip ⟵ flip(coin)
        current_flip ⟵ flip(coin)
return current_flip
```

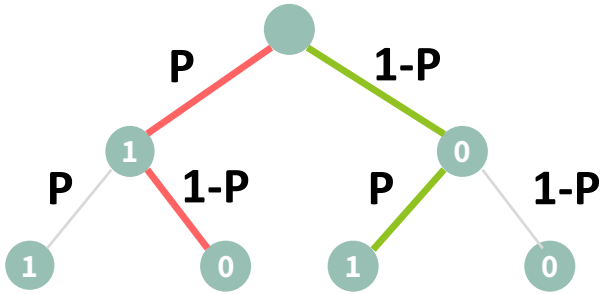
נתון מטבע בעל הסתברות P ל"פלי", כאשר P לא ידועה.

הראו כיצד ליצור מטבע לא מוטה שבו:

$$P(\text{פלי}) = P(\text{עץ}) = \frac{1}{2}$$

תנו סכימה שבה תוחלת מספר ההטלות של המטבע הנתון כדי לקבל הטלה אחת של המטבע הלא מוטה הינה לא יותר מאשר $\frac{1}{P(1-P)}$

רמז: התבוננו בשתי הטלות רצופות של המטבע הנתון.



ניתוח הסתברותי:

נגדיר את המאורע שבמטבע ההוגן יצא 1 (פלי) בתור c. מתקיים:

$$P(c) = \frac{P(\text{יצא 0 ואחריו 1})}{P(\text{שתי תוצאות שונות})} = \frac{P(1-P)}{2P(1-P)} = \frac{1}{2}$$

נשים לב שביצענו **ניסוי ברנולי**, לכן, תוחלת מספר ההטלות שביצענו עד לקבלת הטלה של מטבע הוגן היא **תוחלת משתנה גאומטרי** שבו ההסתברות להצלחה היא ההסתברות לקבל שתי תוצאות שונות בזוג הטלות אחד.

התוחלת למספר הניסויים שנבצע: $\frac{1}{2P(1-P)}$.

מכיוון שכל ניסוי מורכב משתי הטלות של המטבע המקורי – נקבל שתוחלת מספר הטלות המטבע היא $\frac{1}{P(1-P)}$ כנדרש.

לסיכום – מה ראינו הסמסטר - גרפים

מציאת רכיבים קשירים היטב

מציאת עפ"מ

פרים — אלגוריתם חמדן

קרושקל — אלגוריתם חמדן

מציאת מסלולים קצרים ביותר

דייקסטרה — אלגוריתם חמדן

משקלים חיוביים בלבד

בלמן פורד

משקלים ב-R

פתרון מערכת אילוצי הפרשים

פלויד וורשל — תכנון דינמי

מסלולים קצרים ביותר בין כל זוג קדקודים

מציאת זרימה מקסימלית

פורד-פולקרסון

אילוצי קיבול

$f(u,v) \le c(u,v)$

סימטריה נגדית

$f(u,v) = -f(v,u)$

חוק שימור הזרימה

$$\sum_{v \in V} f(u,v) = 0$$

חתך מינימלי – זרימה מקסימלית

לסיכום – מה ראינו הסמסטר- נושאים נוספים

התאמת מחרוזות

רבין קרפ
המרת המחרוזת למספר, נבדוק בדיקה מלאה רק התנגשויות

KMP וחישוב פונקציית הרישא

התאמת מחרוזות באמצעות קונבולוציה

כפל פולינומים

FFT וההתמרה ההפוכה
שימוש בהתמרת פורייה המהירה וההתמרה ההפוכה כדי לכפול פולינומים בצורה יעילה.

אלגוריתמים אקראיים

אלגוריתם לאס וגאס
לדוגמה: מיון מהיר עם בחירת ציר אקראי

אלגוריתם מונטה קרלו
לדוגמה: אלגוריתם קרגר למציאת חתך מינימלי בגרף

בהצלחה בתקופת המבחנים ☺