מבוא למדעי המחשב - סמסטר א' תש"פ

עבודת בית מספר 1

צוות התרגיל: אופיר גגולשוילי, בר סימן טוב, מיכל שמש
תאריך פרסום: 31.10.19
תאריך הגשה: 13.11.19, 12:00 בצהריים
תאריך עדכון אחרון – 9.11.19

הוראות מקדימות:

הגשת עבודות בית

1. קראו את העבודה מתחילתה ועד סופה לפני שאתם מתחילים לפתור אותה. ודאו שאתם מבינים את כל המשימות. רמת הקושי של המשימות אינה אחידה: הפתרון של חלק מהמשימות קל יותר, ואחרות מצריכות חקירה מתמטית - שאותה תוכלו לבצע בספרייה או בעזרת מקורות דרך רשת האינטרנט .בתשובות שבהן אתם מסתמכים על עובדות מתמטיות שלא הוצגו בשיעורים, יש להוסיף כהערה במקום המתאים בקוד את ציטוט העובדה המתמטית ואת המקור (כגון ספר או אתר).
2. עבודה זו תוגש ביחידים. על מנת להגיש את העבודה יש להירשם למערכת ההגשות (Submission System). את הרישום למערכת ההגשות מומלץ לבצע כבר עכשיו, טרם הגשת העבודה (קחו בחשבון כי הגשה באיחור לא מתקבלת). את הגשת העבודה ניתן לבצע רק לאחר הרישום למערכת.
3. לעבודה מצורפים קבצי Java עם שמות כגון Task<n>.java, כאשר <n> מציין את מספר המשימה המתאימה לקובץ (לדוגמא, קובץ Java בשם Task2.java מתאים למשימה מספר 2) . צרו תיקייה חדשה והעתיקו את קבצי ה- Java לתוכה. עליכם לערוך את הקבצים האלו בהתאם למפורט בתרגיל ולהגישם כפתרון, מכווצים כקובץ ZIP יחיד. בנוסף לקבצי ה java, יש להוסיף לתיקייה המכווצת את קובץ ה readme.txt כמפורט בהמשך. שימו לב: עליכם להגיש רק את קבצי ה- Java ואת קובץ ה readme. אין לשנות את שמות הקבצים. אין להגיש קבצים נוספים. שם קובץ ה-ZIP יכול להיות כרצונכם, אך באנגלית בלבד. בנוסף, הקבצים שתגישו יכולים להכיל טקסט המורכב מאותיות באנגלית, מספרים וסימני פיסוק בלבד. טקסט אשר יכיל תווים אחרים (אותיות בעברית, יוונית וכד'..) לא יתקבל. הקפידו לא להשאיר חלקי קוד אשר אינם חלק מהתוכנית (לדוגמה בדיקות שכתבתם עבור עצמכם).
4. קבצים שיוגשו שלא על פי הנחיות אלו לא ייבדקו. את קובץ ה-ZIP יש להגיש ב- Submission System. פרטים בעניין ההרשמה ואיך להגיש את העבודה תוכלו למצוא באתר.

בדיקת עבודות הבית

5. עבודות הבית נבדקות גם באופן ידני וגם באופן אוטומטי. הבדיקה האוטומטית מתייחסת לפלט התכנית המודפס למסך. לכן, יש להקפיד על ההוראות ולבצע אותן במדוייק. כל הדפסה אשר אינה עונה בדיוק על הדרישות המופיעות בעבודה (כולל שורות, רווחים, סימני פיסוק או כל תו אחר - מיותרים, חסרים או מופיעים בסדר שונה מהנדרש), לא תעבור את הבדיקה האוטומטית ולכן תגרור פגיעה בציון.
6. סגנון כתיבת הקוד ייבדק באופן ידני. יש להקפיד על כתיבת קוד יעיל, ברור, על מתן שמות משמעותיים למשתנים, על הזחות (אינדנטציה), ועל הוספת הערות בקוד המסבירות את תפקידם של מקטעי הקוד השונים. אין צורך למלא את הקוד בהערות סתמיות, אך חשוב לכתוב הערות בנקודות קריטיות, המסבירות קטעים חשובים בקוד. הערות יש לרשום אך ורק באנגלית. כתיבת קוד אשר אינה עומדת בדרישות אלו תגרור הפחתה בציון העבודה.

7. בכל פעם שאתם מתבקשים להדפיס למסך, עליכם להשתמש בפונקציה System.out.println(), אשר מדפיסה למסך ויורדת שורה (לכן, כל פעולת הדפסה תופיע בשורה נפרדת). אין להדפיס למסך דברים מיותרים (כגון: "please enter an integer").
8. בכדי לקלוט נתונים מהמשתמש יש להשתמש ב-Scanner, כפי שנלמד בכיתה.

עזרה והנחיה

9. לכל עבודת בית בקורס יש צוות שאחראי לה. ניתן לפנות לצוות בשעות הקבלה. פירוט שמות האחראים לעבודה מופיע במסמך זה וכן באתר הקורס, כמו גם פירוט שעות הקבלה. בתגבור השני של הסמסטר, 3.11.19-6.11.19 נפתור באופן מודרך את משימות 1, 2, 4א. כמו כן, אתם יכולים להיעזר בפורום ולפנות בשאלות לחבריכם לכיתה. צוות הקורס עובר על השאלות ונותן מענה במקרה הצורך.
10. בכל בעיה אישית הקשורה בעבודה (מילואים, אשפוז וכו'), אנא פנו אלינו דרך מערכת הפניות, כפי שמוסבר באתר הקורס.
11. אנחנו ממליצים בחום להעלות פתרון למערכת ההגשה לאחר כל סעיף שפתרתם. הבדיקה תתבצע על הגרסה האחרונה שהועלתה (בלבד!).

הערות ספציפיות לעבודת בית זו

12. בעבודה זו 4 משימות ו- 10 תתי-משימות וסך הנקודות המקסימלי הוא 100. שימו לב שמספר הנקודות לכל תת-משימה אחיד (10 נקודות למשימה) ואינו מצביע על קושי המשימה.
13. בעבודה זו מותר להשתמש בידע שנלמד עד הרצאה 3 (כולל), וכן עד תרגול 2 (כולל). לא ניתן להשתמש במערכים, מחרוזות, פונקציות, או כל צורת קוד אחרת אשר לא נלמדה בכיתה.
14. בעבודה זו, בתוכניות אותן אתם מגישים, כל המשתנים עבור מספרים שלמים צריכים להיות מטיפוס **int.**
15. בכל המשימות ניתן להניח כי הקלט תקין.

יושר אקדמי

הימנעו מהעתקות! ההגשה היא ביחידים. אם מוגשות שתי עבודות עם קוד זהה או אפילו דומה - זוהי העתקה, אשר תדווח לאלתר לוועדת משמעת. אם טרם עיינתם בסילבוס הקורס אנא עשו זאת כעת.

מומלץ לקרוא היטב את כל ההוראות המקדימות ורק לאחר מכן להתחיל בפתרון המשימות. ודאו שאתם יודעים לפתוח קבוצת הגשה (עבור עצמכם) במערכת ההגשות.

הצהרה (0 נקודות)

פתחו את הקובץ readme.txt וכיתבו בו את שמכם ומספר תעודת הזהות שלכם. משמעות פעולה זו היא שאתם מסכימים על הכתוב בו. דוגמה:

I, Israel Israeli (123456789), assert that the work I submitted is entirely my own.
I have not received any part from any other student in the class, nor did I give parts of it for use to others.
I realize that if my work is found to contain code that is not originally my own, a formal case will be opened against me with the BGU disciplinary committee.

יש לצרף את הקובץ **readme.txt** לקובץ ה- **zip** שמכיל את קבצי ה-**Task<n>.java** .

הקדמה: חילוק שלמים ושארית חלוקה

לכל שני מספרים שלמים a, b כך ש- $b \neq 0$, החלק השלם במנה $\frac{a}{b}$ הוא מספר שלם q כך ש- $a = q \cdot b + r$ ו- $r < b$ שלם. המספר r נקרא שארית החלוקה של a ב- b ומסומנת $r = a\%b$.

למשל בעבור $a = 13$ ו- $b = 3$ החלק השלם במנה $\frac{13}{3}$ הוא $q = 4$ ושארית החלוקה $r = 1$ כיוון שמתקיים $13 = 4 \cdot 3 + 1$.

משימה 1 - משימת חימום

פתחו את הקובץ Task1.java וכתבו בו תכנית אשר קולטת מהמשתמש ארבעה מספרים שלמים a, b, q, r ומדפיסה למסך "yes" אם $r < b$, $b \neq 0$ ו- $a = q \cdot b + r$ ו- "no" אחרת. שימו לב, יש לכתוב באותיות קטנות, ללא רווחים או סימנים כלשהם.
המספרים ייקלטו בסדר הבא (משמאל לימין): a, b, q, r.
ניתן להניח כי הקלט תקין, כלומר כי a, b, q, r הם מספרים שלמים.

דוגמאות:
אם הקלט הוא $a = 10,\ b = 4,\ q = 2,\ r = 1$ אזי הפלט יהיה:
`no`
אם הקלט הוא $a = 10,\ b = 4,\ q = 2,\ r = 2$ אזי הפלט יהיה:
`yes`
אם הקלט הוא $a = 9,\ b = 3,\ q = 3,\ r = 0$ אזי הפלט יהיה:
`yes`
אם הקלט הוא $a = 5,\ b = 7,\ q = 0,\ r = 5$ אזי הפלט יהיה:
`yes`
להזכירכם, אין להדפיס דברים מיותרים למסך (כגון: "please enter four integers").

סיימתם חלק זה? כל הכבוד! העלו את הגרסה האחרונה של עבודתם למערכת ההגשה.

משימה 2 – עוד משימת חימום

פתחו את הקובץ Task2.java וכתבו בו תכנית אשר קולטת מהמשתמש שני מספרים שלמים a, b כך ש $a < b$ ומגרילה באקראי מספר שלם n בתחום $[a, b]$. במילים: המספר n שיוגרל צריך לקיים $a \leq n \leq b$.

הדרכה: יש להשתמש בפקודה $Math.random()$ המחזירה מספר אקראי x בתחום החצי פתוח $[0,1)$. במילים: x שמוחזר ע"י הפקודה מקיים $0 \leq x < 1$. המספרים ייקלטו בסדר הבא (משמאל לימין): a, b.
ניתן להניח כי הקלט תקין, כלומר כי a, b הם מספרים שלמים וכן כי $a < b$.

דוגמאות:
אם הקלט הוא $a = 2,\ b = 24$ אזי פלט אפשרי יכול להיות:
`17`
אם הקלט הוא $a = -4,\ b = 5$ אזי פלט אפשרי יכול להיות:
`-4`

סיימתם חלק זה? כל הכבוד! העלו את הגרסה האחרונה של עבודתם למערכת ההגשה.

סימונים:

- MV מייצג את הערך המתקבל מהפקודה `int MV = Integer.MAX_VALUE;`

משימה 3: חזקות של 2 ושארית חלוקה

משימה 3א:

פתחו את הקובץ Task3a וכתבו בו תכנית אשר קולטת מהמשתמש מספר שלם אי-שלילי n ומדפיסה למסך את הערך 2^n.
זיכרו כי יש להשתמש במשתנים מטיפוס int בלבד. על התוכנית לחשב נכונה את החזקות של 2 עבור כל ערך של n בין 0 ל- 30 כולל.

דוגמאות:

אם הקלט הוא $n = 0$ אז התכנית תדפיס:

1

אם הקלט הוא $n = 1$ אז התכנית תדפיס:

2

אם הקלט הוא $n = 10$ אז התכנית תדפיס:

1024

אם הקלט הוא $n = 31$ אז התכנית תדפיס:

-2147483648

נסו להבין מדוע.

סיימתם חלק זה? כל הכבוד! העלו את הגרסה האחרונה של עבודתם למערכת ההגשה.

שימו לב: בחלק זה אין להשתמש בספריה Math. עליכם לחשב את 2^n ע"י שימוש בלולאה.
ניתן להניח כי הקלט תקין, כלומר כי n הוא שלם אי-שלילי בין 0 ל- 30 כולל.

משימה 3ב:

פתחו את הקובץ Task3b.java וכתבו בו תכנית אשר קולטת מהמשתמש שני מספרים שלמים n, k ומדפיסה למסך את הערך של $2^n \% k$, כלומר את שארית החלוקה של 2^n ב-k.
המספרים ייקלטו בסדר הבא (משמאל לימין): n, k.

דוגמאות:

אם ערכי הקלט הם $n = 10,\ k = 54$ אזי הפלט יהיה:

52

כיוון ש- $2^{10} = 1024$ ומתקיים ש- $2^{10} = 54 * 18 + 52$

אם בקלט שני הערכים הם $n = 35,\ k = 151$ אזי הפלט יהיה:

32

כיוון ש- $2^{35} = 34{,}359{,}738{,}368$ ומתקיים ש- $2^{35} = 151 * 227{,}547{,}936 + 32$

סיימתם חלק זה? כל הכבוד! העלו את הגרסה האחרונה של עבודתם למערכת ההגשה.

יש להניח כי המספרים n, k הם שלמים אי-שליליים וכי $1 < k < \sqrt{MV}$.
על התכנית לחשב נכונה את $2^n \% k$ לכל ערך כנ"ל של n ו-k (בפרט עבור $n = 31$).

הדרכת חובה: על מנת לפתור נכונה תרגיל זה גם עבור ערכים גדולים של n, יש להשתמש בעובדה הבאה:

$$(a \cdot b)\%k = \big((a\%k) \cdot (b\%k)\big)\ \%k.$$

לדוגמה: $(6 \cdot 7)\%5 = 2 = \big((6\%5) \cdot (7\%5)\big)\ \%5$

משימה 4: בדיקת ראשוניות של מספר

תזכורת :

מספר ראשוני (p prime) הוא מספר שלם גדול מ-1 אשר מתחלק ללא שארית רק ב-1 ובעצמו. לדוגמה: 2,3,5,7,…

מספר פריק (composite) הוא מספר שלם אשר קיים לו מחלק שלם גדול מ-1 השונה מ-1 ומעצמו. לדוגמה: 4,6,8,9,…

הנחיה: בכל חלקי המשימה הבאים אין להשתמש בפונקציה $Math.pow$ (למעט עבור בדיקות נכונות אשר תכתבו בעצמכם ואינן כלולות בקוד המוגש).

משימה 4א: אלגוריתם נאיבי לבדיקת ראשוניות של מספר

פתחו את הקובץ Task4a.java וכתבו בו תכנית אשר קולטת מהמשתמש מספר שלם $1 < n \leq MV$ ומדפיסה למסך "prime" אם n ראשוני ו- "composite" אחרת.

הדרכת חובה: יש לבדוק בלולאה האם קיים ל- n מחלק שאינו טריוויאלי.

דוגמאות:

אם הקלט הוא $n = 10$ אזי הפלט יהיה:

composite

אם הקלט הוא $n = 11$ אזי הפלט יהיה:

prime

ניתן להניח כי הקלט תקין, כלומר כי n הוא שלם $1 < n \leq MV$.

מספר הראשוניים

לכל מספר שלם $n > 1$ נסמן ב- $\pi(n)$ את מספר המספרים הראשוניים אשר קטנים או שווים ל- n.

לדוגמה: $\pi(2) = 1,\ \pi(5) = 3,\ \pi(20) = 8$

משימה 4ב: אלגוריתם נאיבי לבדיקת מספר ראשוניים

פתחו את הקובץ Task4b.java וכתבו בו תכנית אשר קולטת מהמשתמש מספר שלם n ומדפיסה למסך את $\pi(n)$.

דוגמאות:

אם הקלט הוא $n = -10$ אז התכנית תדפיס:

0

אם הקלט הוא $n = 0$ אז התכנית תדפיס:

0

אם הקלט הוא $n = 2$ אז התכנית תדפיס:

1

אם הקלט הוא $n = 5$ אז התכנית תדפיס:

3

אם הקלט הוא $n = 20$ אז התכנית תדפיס:

8

ניתן להניח כי הקלט תקין, כלומר כי n הוא מספר שלם ו- $n \leq MV$.

סיימתם חלק זה ?כל הכבוד !העלו את הגרסה האחרונה של עבודתם למערכת ההגשה.

אלגוריתם מילר-רבין – מבוא

האלגוריתם הנאיבי לבדיקת ראשוניות מספר נתון מסוגל לספק תשובה נכונה בזמן סביר עבור מספרים שאינם מאוד גדולים. כאשר מדובר במספרים גדולים (בעלי 200 ספרות, לדוגמא) האלגוריתם הנאיבי ירוץ בזמן ארוך מדיי ולא נוכל לקבל תשובה בזמן סביר.

אחד האלגוריתמים הנפוצים והמקובלים כיום לבדיקת ראשוניות של מספר גדול הוא האלגוריתם של מילר-רבין.

אלגוריתם זה הוא אלגוריתם אקראי, דבר הגורר אפשרות שתוחזר תשובה שגויה (בהמשך ישנו פירוט לגבי הסיכוי לשגיאה שלו).

בתרגיל מודרך זה (בו כל סעיף מסתמך על סעיפים קודמים) נבדוק האם מספר נתון הוא ראשוני. הבדיקה תיעשה בהתאם לאלגוריתם מילר-רבין.

בשלב ראשון (משימות ג'-ה') נקלוט מהמשתמש מספר אי-זוגי n ונבצע בדיקה בודדת עבור ראשוניות המספר n שהגרלנו עם הסתברות להחזרת תשובה שגויה שלא עולה על ¼. בבדיקה זו נבדוק האם מספר b בתחום $[2, n-1]$ שהוגרל באקראי עומד בתנאי מסויים שהוגדר ע"י מילר-רבין (אותו נציג בהמשך) ביחס ל- n , נסמנו (*). למספר b בתחום $[2, n-1]$ אשר עומד בתנאי (*) עבור n נקרא עד (witness), כיוון שהוא מעיד על כך ש-n מספר פריק.

בשלב שני (משימה ו') נחזור על הבדיקה באופן בלתי תלוי k פעמים ובכך נקטין את ההסתברות לשגיאה כך שלא תעלה על $\left(\frac{1}{4}\right)^k$. נשים לב כי עבור הערך $k = 50$ אנו מקטינים את ההסתברות לשגיאה ל- $\frac{1}{2^{100}} = \frac{1}{4^{50}}$.

סדר הפעולות (האלגוריתם) בשלב הראשון יהיה כדלהלן:

1. נקלוט מהמשתמש מספר אי-זוגי $1 < n$.
2. נגריל מספר שלם יחיד b בתחום $[2, n-1]$.
3. נבדוק האם b שהגרלנו עומד בתנאי (*) ביחס ל- n.
4. אם התנאי (*) מתקיים, נכריז כי n פריק.
5. אחרת, נכריז כי n ראשוני.

עובדות מתמטיות:

- אם n פריק, ישנם לפחות $\frac{3}{4}n$ מספרים b בתחום $[2, n-1]$ אשר יעמדו בתנאי (*) ולכן ההסתברות להגריל באקראי b העומד בתנאי גדולה או שווה ל- ¾.
- אם n פריק, ישנם פחות מ- $\frac{1}{4}n$ מספרים b בתחום $[2, n-1]$ אשר לא יעמדו בתנאי (*) ולכן ההסתברות להגריל באקראי b שלא עומד בתנאי קטנה מ-¼.
- אם n ראשוני אז כל b בתחום $[2, n-1]$ לא עומד בתנאי (*).

לאור עובדות אלו נוכל להבין את האלגוריתם באופן הבא:

1. אם הכרזנו כי n פריק, אזי אנו יודעים בוודאות כי הוא פריק.
2. אם הכרזנו כי n ראשוני, קיימים שני מצבים אפשריים:
   - n ראשוני. אז החזרנו תשובה נכונה.
   - n פריק. אז טעינו: מכיוון שהוגרל מספר b אשר לא עמד בתנאי (*).

נצא לדרך!

משימה 4ג:

פתחו את הקובץ Task4c.java וכתבו בו תכנית אשר קולטת מהמשתמש מספר אי זוגי $1 < n < \sqrt{MV}$, מגרילה מספר שלם b בתחום $[2, n-1]$ ומדפיסה אותו למסך.

ניתן להניח כי הקלט תקין, כלומר כי n הוא שלם אי זוגי כך ש- $1 < n < \sqrt{MV}$.

דוגמא:

אם הקלט הוא $n = 22{,}317$ אזי פלט אפשרי הוא:

1684

משימה 4ד:

בהינתן מספר $x > 1$ זוגי ניתן ליצגו באופן הבא $x = 2^s \cdot d$ כך ש- $s > 0$ ו- d אי-זוגי.

דוגמאות: $6 = 2^1 \cdot 3, 60 = 2^2 \cdot 15$

פתחו את הקובץ Task4d.java וכתבו בו תכנית אשר קולטת מהמשתמש מספר אי-זוגי $1 < n < \sqrt{MV}$ ומוצאת את s, d כך ש- $n - 1 = 2^s \cdot d$, $s > 0$ ו- d אי-זוגי. התוכנית תדפיס אותם למסך, כל אחד בשורה נפרדת, באופן הבא: המספר הראשון שיודפס יהיה s. המספר השני שיודפס יהיה d.

ניתן להניח כי הקלט תקין, כלומר כי n הוא שלם אי זוגי כך ש- $1 < n < \sqrt{MV}$.

<u>הדרכת חובה</u>: את החישוב של s בצעו ע"י שימוש בלולאה.

דוגמא:

בהנחה והקלט הוא המספר 12,317 הפלט יהיה:

2

3079

כיוון ש- $12{,}317 - 1 = 2^2 \cdot 3{,}079$.

סיימתם חלק זה ?כל הכבוד !העלו את הגרסה האחרונה של עבודתם למערכת ההגשה.

סימון: שקילות מודולו n

יהיו $n > 1$ מספר טבעי ו- a מספר טבעי. אם שארית החלוקה של a ב- n היא b (כלומר $a\%n = b$) אז נסמן כי $a \equiv b \; mod(n)$ (נאמר ש- a ו- b שקולים מודולו n).

דוגמאות: $4 \equiv 1 \; mod(3), \; 13 \equiv 6 \; mod(7)$

משימה 4ה:

פתחו את הקובץ Task4e.java וכתבו בו תכנית אשר קולטת מהמשתמש 4 מספרים n, b, s, d בסדר הזה משמאל לימין כך ש:

- n מספר אי-זוגי בתחום $[2, \sqrt{MV}]$
- b מספר שלם בתחום $[2, n-1]$
- s, d שלמים כך ש- $n-1 = 2^s \cdot d$, $s > 0$, ו- d אי-זוגי

ובודקת האם b עומד בתנאי הבא, אשר נסמנו ב- (*) עבור n:

תנאי (*) מציין כי:

1. לא מתקיים $b^d \equiv 1 \; mod(n)$

וגם

2. לכל $0 \le i \le s-1$ לא מתקיים:
   - $b^{2^i \cdot d} \equiv (n-1) \; mod(n)$

כלומר, <u>כל</u> $s+1$ השקילויות לעיל <u>לא מתקיימות</u>.

אם התנאי מתקיים התוכנית תדפיס

b is a witness. n is composite.

אחרת התוכנית תדפיס

We assume n is prime.

ניתן להניח כי הקלט תקין, כלומר כי:

- n מספר אי-זוגי בתחום $[2, \sqrt{MV}]$
- b מספר שלם בתחום $[2, n-1]$
- s, d שלמים כך ש- $n-1 = 2^s \cdot d$, $s > 0$, ו- d אי-זוגי

דוגמאות:

אם ערכי הקלט הם $n = 25{,}123$, $b = 11{,}309$, $s = 1$, $d = 12{,}561$ אזי הפלט יהיה:

11309 is a witness. 25123 is composite.

אם ערכי הקלט הם $n = 31{,}663$, $b = 16{,}116$, $s = 1$, $d = 15{,}831$ אזי הפלט יהיה:

We assume 31663 is prime.

(זאת כיוון ש- $16{,}116^{15{,}831} \; mod(31{,}663) = 31{,}662$)

בשתי הדוגמאות לא ניתן לייצג במשתנה מסוג int את הערך $b^{2^i \cdot d}$ אפילו עבור ערכי i קטנים. על הפתרון שלכם לעבוד גם עבור מצבים אלו.

רמז: ראו הדרכת חובה במשימה 3ב.

סיימתם חלק זה ?כל הכבוד !העלו את הגרסה האחרונה של עבודתם למערכת ההגשה.

משימה 4ו:

פתחו את הקובץ Task4f.java וכתבו בו תכנית אשר קולטת מהמשתמש 4 מספרים n, s, d, k בסדר הזה משמאל לימין כך ש-

- n מספר אי-זוגי בתחום $[2, \sqrt{MV}]$
- s, d שלמים כך ש- $s > 0,\ n - 1 = 2^s \cdot d$, ו- d אי-זוגי
- k שלם כך ש- $1 < k < 51$

ופועלת באופן הבא:

התוכנית חוזרת על הפעולה הבאה k פעמים:

- התוכנית מגרילה מספר b שלם בתחום $[2, n-1]$ ובודקת האם b עומד בתנאי (*) עבור n.
- אם באחת ההגרלות הוגרל מספר b אשר עמד בתנאי (*) התוכנית תדפיס

b is a witness. n is composite.

- אחרת, התוכנית תדפיס

We assume n is prime.

שימו לב: אין חובה לבצע את כל k ההגרלות במידה והוגרל מספר b אשר עמד בתנאי (*) עבור n.

ניתן להניח כי הקלט תקין, כלומר כי:

- n מספר אי-זוגי בתחום $[2, \sqrt{MV}]$
- s, d שלמים כך ש- $s > 0,\ n - 1 = 2^s \cdot d$, ו- d אי-זוגי
- k שלם כך ש- $1 < k < 51$

סיימתם חלק זה ?כל הכבוד !העלו את הגרסה האחרונה של עבודתם למערכת ההגשה.

דוגמאות:

אם ערכי הקלט הם $n = 4{,}793\ ,\ s = 3,\ d = 599,\ k = 10$ אזי הפלט יהיה:

We assume 4793 is prime.

אם ערכי הקלט הם $n = 15{,}379\ , s = 1, d = 7689,\ k = 15$ אזי פלט אפשרי יהיה:

4191 is a witness. 15379 is composite.

<u>הערה</u>: בתוכנית זו, ההסתברות לשגיאה אינה עולה על $\left(\frac{1}{4}\right)^k$.

במילים אחרות: אם נקלט מספר n פריק, ההסתברות שנצהיר כי הוא ראשוני לא עולה על $\left(\frac{1}{4}\right)^k$.

אם נקלט מספר n ראשוני, כל b שיוגרל לא יעמוד בתנאי (*) עבר n ולכן נצהיר בוודאות כי הוא ראשוני.

סטודנטים המתעניינים בקריאה נוספת מוזמנים לקרוא על המושגים הבאים (בהם תיתקלו במהלך לימודיכם):

1. מספר ראשוני – Prime Number
2. פירוק מספר לגורמים ראשוניים – Finding Factors of a Number
3. חשבון מודולרי (חשבון קונגרואנציות) – Modular Arithmetic
4. המשפט הקטן של פרמה – Fermat's Little Theorem
5. אלגוריתם דטרמיניסטי – Deterministic Algorithm
6. אלגוריתם אקראי – Randomized Algorithm
7. צפיפות הראשוניים

בהצלחה !