מחשוב זמן אמת

תרגיל מספר 1

התרגיל הזה מנועד להמחיש אלגוריתם דטרמיניסטי.

הבעיה שבאמצעותו נעשה זאת יהיה בעיית בדיקה האם מספר הוא ראשוני או לא.

בהינתן מספר שלם הדרך הפשוטה ביותר היא לבדוק אם כל המספרים מ-2 עד אינם מחלקים את המספר. אבל ניתן לשפר את האלגוריתם לכדי חצי ע"י הטכניקה:

* בדוק אם המספר הוא 2. אזי הוא ראשוני.
* בדוק אם המספר הוא זוגי ולא 2, במקרה הזה המספר אינו ראשוני ו-2 מחלק אותו.
* בדוק עבור המספרים 3, 6, 7, 9, 11, … , אם המספר ראשוני.

אנחנו נממש גרסה שלוקחת את הרעיון קצת הלאה, מוריד את מספר הבדיקות לכדי שליש:

* בדוק אם המספר הוא 2. אזי הוא ראשוני.
* בדוק אם המספר הוא 3. אזי הוא ראשוני.
* בדוק אם המספר הוא זוגי ולא 2, במקרה הזה המספר אינו ראשוני ו-2 מחלק אותו.
* בדוק אם המספר מתחלק ב-3 ואיננו 3, במקרה הזה המספר אינו ראשוני ו-3 מחלק אותו.
* בדוק במקביל עבור 2 סדרות המספרים

5, 11, 17, 23, 29, … ,

7, 13, 19, 25, 31, … ,

אם המספר ראשוני.

האלגוריתם הסטנדרטי בדרך כלל נעצר כאשר מתגלה שהמספר אינו ראשוני, אבל דווקא בהקשר שלנו איננו מעוניינים בחסכון בחישובים הפוטנציאלי הזה.

עליך לממש ארבע גרסאות של האלגוריתם:

1. גרסה סטנדרטית שעוצרת אם המספר אינו ראשוני.
2. גרסה דטרמיניסטית שעושה בכל מקרה את כל הבדיקות אבל מדווחת את המקרה הראשון של אי התאמה.
3. גרסה סטנדרטית שסופרת את מספר הבדיקות ומדפיסה אותם להראות שהאלגוריתם אינו דטרמיניסטי כאשר המספר אינו ראשוני.
4. גרסה דטרמיניסטית שסופרת את מספר הבדיקות ומדפיסה אותם להראות שהאלגוריתם אכן דטרמיניסטי.

התוכנית הראשית הבאה בודקת את שני הראשונים:

**int main(void)**

**{**

**unsigned long long int n, k;**

**int is\_prime;**

**printf("Enter unsigned integer:\n");**

**scanf("%Lu", &n);**

**system("time");**

**is\_prime = test\_prime(n,&k);**

**system("time");**

**if (is\_prime == 1)**

**printf("%Lu IS a prime\n", n);**

**else**

**{**

**printf("%Lu is NOT a prime\n", n);**

**printf("%Lu = %Lu \* %Lu\n", n, k, n/k);**

**} /\* else \*/**

**return 0;**

**} /\* main \*/**

התוכנית הראשית הבאה בודקת את הגרסה הסופרת את מספר ההשוואות

**int main(void)**

**{**

**unsigned long long int n, k;**

**int is\_prime;**

**unsigned long count;**

**printf("Enter unsigned integer:\n");**

**scanf("%Lu", &n);**

**system("time");**

**is\_prime = test\_prime(n,&k, &count );**

**system("time");**

**if (is\_prime == 1)**

**printf("%Lu IS a prime\n", n);**

**else**

**{**

**printf("%Lu is NOT a prime\n", n);**

**printf("%Lu = %Lu \* %Lu\n", n, k, n/k);**

**} /\* else \*/**

**printf("count = %lu\n", count);**

**return 0;**

**} /\* main \*/**

דוגמאות ריצה:

1. 2.

**Enter unsigned integer:**

**1990232731**

**The current time is: 20:07:14.32**

**Enter the new time:**

**The current time is: 20:07:14.98**

**Enter the new time:**

**1990232731 is NOT a prime**

**1990232731 = 5179 \* 384289**

**תוכנית 3:**

**Enter unsigned integer:**

**1990232731**

**The current time is: 20:10:00.48**

**Enter the new time:**

**The current time is: 20:10:01.60**

**Enter the new time:**

**1990232731 is NOT a prime**

**1990232731 = 5179 \* 384289**

**count = 1729**

**Enter unsigned integer:**

**1990232743**

**The current time is: 20:11:31.36**

**Enter the new time:**

**The current time is: 20:11:31.98**

**Enter the new time:**

**1990232743 IS a prime**

**count = 14873**

**תוכנית 4:**

**Enter unsigned integer:**

**1990232743**

**The current time is: 20:13:37.77**

**Enter the new time:**

**The current time is: 20:13:38.54**

**Enter the new time:**

**1990232743 IS a prime**

**count = 14873**

**D:\ronn\RealTimeComputing\targilim\targilim26a\targ1>a**

**Enter unsigned integer:**

**1990232731**

**The current time is: 20:13:57.33**

**Enter the new time:**

**The current time is: 20:13:57.97**

**Enter the new time:**

**1990232731 is NOT a prime**

**1990232731 = 5179 \* 384289**

**count = 14873**