# ממן 15

אלגוריתמיקה 20290

06.09.2019

## 

[**ממן**](#_6wpwt0gra8n2) **15 1**

[שאלה](#_940e35q6racc) 1 2

[סעיף א](#_31ovr3cl7qcs) 2

[סעיף ב](#_68vudqdj6zn2) 2

[שאלה](#_mmwybknx364v) 2 3

[סעיף א](#_gmbvvdxh9ghk) 3

[סעיף ב](#_t8xd58u39cyn) 3

[השוואה בין הסעיפים](#_3sj89o2zd9fi) 3

[שאלה](#_9ijxdb4k40o) 3 4

[הפרוטוקול](#_p5qp223uhwdh) 4

[נכונות](#_dtyjiregdmll) 4

[שאלה](#_e4kgdiej4dkn) 4 5

[סעיף א](#_72ucdni4ohtf) 5

[סעיף ב](#_iw0kusj9cdrv) 5

[סעיף ג](#_sdhs5xcsac6o) 5

[סעיף ד](#_z6he2n1zywd3) 5

[סעיף ה](#_d6uaettvkbo5) 5

[שאלה](#_zgtb93v9pdid) 5 6

[סעיף ב](#_7y43vvljeq5z) 6

[סעיף ג](#_elnmtecn84sv) 6

[סעיף ד](#_tvgmr9qzjcux) 6

[סעיף ה](#_x03rhwd389bq) 6

[שאלה](#_gezy189lgs2t) 6 7

## 

## שאלה 1

### סעיף א

הגרסה המקבילית שלי למיון מהיר מתבססת על ההאלגוריתם המוצג בעמ' 96 במדריך הלמידה.

את שלב החלוקה לא ניתן למקבל ביעילות שכן לא ידוע איזה חלק של המערך כל מעבד יכסה, לאחר שלב החלוקה הראשון, ניתן למקבל את הקריאות הרקורסיביות בשלבים 2.2 ו- 2.3, כך כל מעבד יקבל חצי מהמערך וימשיך את העבודה עליו במקביל.

### סעיף ב

בכל איטרציה מספר המעבדים שבשימוש מוכפל, לכן לאחר log n שלבים (מספר הפעמים שנחלק את n לשתיים במקרה הטוב) יהיה צורך ב- n מעבדים.  
כמו כן, מתבצע מעבר לינארי ולאחריו חלוקה לשתיים, לכן מתקבלת הסדרה שכידוע מייצגת זמן של .

סיבוכיות המכפלה של המקרה הטוב היא , ונראה כי זוהי אינה סיבוכיות אופטימלית שכן סיבוכיות הגרסה המקורית במקרה הטוב היא .

## שאלה 2

### סעיף א

כל מעבד יעבור לינארית על משכורות ויספור את מספר המשכורות הנמוכות משכר מינימום, וירשום אותו במערך עזר באורך מספר המעבדים, שלב זה מתבצע ב- .

לאחר מכן יתבצע "טורניר", מכל זוג מעבדים סמוכים אחד מהם יסכום את הספירות של שניהם, שלב זה לוקח זמן לוגריתמי ביחס למספר המעבדים, כלומר , לכן לא מגדיל את זמן הריצה.

סיבוכיות המכפלה היא כמובן .

### סעיף ב

כל מעבד יעבור לינארית על משכורות ויספור את מספר המשכורות הנמוכות משכר מינימום, שלב זה מתבצע ב- .

לאחר מכן יתבצע "טורניר", מכל זוג מעבדים סמוכים אחד מהם יסכום את הספירות של שניהם, שלב זה לוקח זמן לוגריתמי ביחס למספר המעבדים, כלומר , לכן לא מגדיל את זמן הריצה (כי לכן )

סיבוכיות המכפלה היא .

### השוואה בין הסעיפים

זמן הריצה של הפתרון מסעיף ב' קצר יותר, אך סיבוכיות המקום שלו גדולה יותר גם כן (מערך העזר הנדרש גדול יותר).

מכיוון שעלות הזיכרון הנדרש לבעיה כזו זניחה יחסית לעלות העיבוד (במחשבים בני ימינו), תחת ההנחה שיש לנו מעבדים, אניח שאין בעיית זיכרון, לכן הפתרון מסעיף ב' הוא עדיף.

במקרה הנוכחי, אשתמש בהערכת גג גסה מאוד כי יש במערכת החינוך 200 אלף עובדים (בחיפוש זריז ברשת מצאתי נתון של 137,567 עובדים בשנת 2013).

כלומר, על מנת לממש את סעיף א' יידרשו 448 מעבדים, ועל מנת לממש את סעיף ב' יידרשו 11,358 מעבדים.

ההבדל בזמן יהיה בין 448 יח' זמן עיבוד לבין 18 יח' זמן עיבוד. במעבד בין ימינו מתבצעות מיליארדי פעולות בשנייה, לכן שני המספרים זניחים לחלוטין.

מכאן שסביר יותר להשתמש בפתרון של סעיף א', עבורו נדרש מספר "סביר" יותר של מעבדים (זהו אכן מספר סביר למחשב ממשלתי לצורך העניין, רק שמגוחך יהיה להשתמש במחשב כזה לחישוב כזה).

## שאלה 3

### הפרוטוקול

אציג את הפרוטוקול החדש עבור מעבד P1 בגרסה זו של הבעיה, הפתרון עבור מעבדים P2 ו- P3 סימטרי לחלוטין.

|  |
| --- |
| P1 protocol:   1. while true:    1. do private work    2. x1 ← yes    3. z ← 1    4. wait until z=2 or z=3 or x2=no or x3=no    5. do critical section    6. x1 ← no |

### נכונות

נראה כי בכל מצב בו P1 רוצה להיכנס לקטע הקריטי לא יהיו בו שלושת המעבדים:

אם x2 או x3 הם "לא", אז P1 יכול להיכנס כי P2 או P3 (בהתאמה) לא בקטע הקריטי.

אם x2 וגם x3 הם "כן" אז אם z הוא 1, P1 לא ייכנס, אם z שונה מאחד, אז z שווה 2 או 3, כלומר מעבד זה 2 או 3 (בהתאמה) מחכה להיכנס לקטע הקריטי אבל עדיין לא תורו, ו- P1 יכול להיכנס.

## שאלה 4

### סעיף א

הרעיון עליו מבוסס האלגוריתם הוא שחיפוש בינארי מתבסס על כך שהמערך ממוין בסדר עולה, לכן אם המערך לא ממוין, חיפוש בינארי לא יעבוד, ומכיוון שבחרנו איבר מתוך המערך אנחנו יודעים שהוא נמצא בו, לכן אם החיפוש הבינארי לא מצא אותו, כנראה שהמערך לא ממוין.

### סעיף ב

האלגוריתם עלול להחזיר תשובה שגויה- שהמערך ממוין גם אם הוא אינו כזה, זה במצב שבמקרה נבחר איבר שהוא אחד מהאיברים המחלקים (כלומר איבר שהוא אמצע של קטע שנבדק).

אם המערך אכן ממוין אז האלגוריתם תמיד ימצא את האיבר, ולכן תמיד יחזיר תשובה נכונה.

כלומר זהו אלגוריתם ממשפחת co-RP.

### סעיף ג

זמן הריצה של חיפוש בינארי הוא log n, כל שאר הפעולות נעשות בזמן קבוע.

אלגוריתם דטרמיניסטי לפתרון בעיה זו יצטרך לבדוק שכל איבר נמצא במקום הנכון, כלומר זמן הריצה שלו יהיה לפחות לינארי. ברור שאלגוריתם נאיבי שבודק את שכל איבר גדול מקודמו ירוץ בזמן לינארי, לכן זהו הזמן האופטימלי.

אכן זמן ריצה של log n הוא "בסדר גדול" מהיר יותר מזמן ריצה לינארי.

### סעיף ד

אם האלגוריתם יבחר את האינדקס 1, כלומר את המספר 2, הוא אכן ימצא אותו (כי בבדיקה מול 3 ולאחר מכן בבדיקה מול 5, האיבר 2 אכן נמצא בצד הנכון).

אם האלגוריתם יבחר את האינדקס 2, כלומר את המספר 5, הוא לא ימצא אותו (כי בבדיקה מול 3 הוא בצד הלא נכון).

אינדקס 3, כלומר מספר 4, לא ימצא.

אינדקס 4, כלומר מספר 3, כן ימצא.

אינדקס 5, כלומר מספר 8, לא ימצא (בבדיקה השנייה מול 7 הוא לא בצד הנכון).

אינדקס 6, כלומר מספר 7, כן ימצא.

אינדקס 7 כלומר מספר 6 לא ימצא (בבדיקה השנייה מול 7 הוא לא בצד הנכון).

הסיכוי של האלג' להחזיר תשובה שגויה הוא הסיכוי שהוא יבחר מספר שהוא יצליח למצוא, כלומר אינדקס 1, 3 או 7, לכן הסיכוי הוא 3/7.

### סעיף ה

על ידי הרצה חוזרת של האלגוריתם שוב ושוב, ניתן להקטין את הסיכוי לטעות. הסיכוי שהאלג' יטעה k פעמים ברציפות הוא , לכן ניתן להקטין את הסיכוי כרצוננו על ידי הגדלת מספר הריצות.

## שאלה 5

### סעיף ב

האדם שאינו מומחה ליין הוא המאמת, והמומחה ליין הוא המוכיח.

המאמת ימזוג 5 כוסות מכל בקבוק (בסתר מהמוכיח, תוך שמירת על רישום מאיזה בקבוק נמזגה כל כוס).

על המוכיח יהיה לזהות את 5 הכוסות שנמזגו לאותו בקבוק.

אם המוכיח יצליח לזהות, המאמת יידע כי המוכיח דובר אמת (הבקבוקים שונים).

אם הבקבוקים שונים, הסיכוי של המומחה לנחש נכון קטן מאוד (כפי שנראה בסעיף ד').

### סעיף ג

אם באמת בקבוקי היין שונים, אז המומחה יצליח לזהות את חמשת הכוסות (על ידי טעימות או על ידי כל אמצעי אחר הבודק שוני בין היינות).

### סעיף ד

ישנן אפשרויות לבחור 5 כוסות מתוך 10, לכן אם הסיכוי של המאמת לנחש בדיוק את חמשת הכוסות הוא 1 ל-252, כלומר 0.0039

### סעיף ה

המאמת לא לומד איך המוכיח זיהה את חמשת הכוסות, כלומר איך המוכיח "פענח" את המידע, לכן זהו פרוטוקול אפס-ידע.

## שאלה 6

השחקן הראשון יבחר את המשבצת האמצעית, לאחר מכן, ישלים שלישייה ביחס למרכז עם כל צעד של היריב.

נסמן לוח:

|  |  |  |
| --- | --- | --- |
| 3 | 2 | 1 |
| 6 | 5 | 4 |
| 9 | 8 | 7 |

השחקן הראשון מניח במשבצת 5.

אם השחקן השני מניח במשבצת 4, השחקן הראשון יניח במשבצת 6.

אם השחקן השני מניח במשבצת 9, השחקן הראשון יניח במשבצת 1.

כך, כדי שהשחקן שהתחיל ישלים את שלישייה 123 למשל, על השחקן השני להניח את 789 לפניו, ולכן יפסיד.