

סיבוכיות מקום

ראינו שחשוב למדוד כמה אלגוריתם נתון הוא יעיל. עד עכשיו המדד שלנו לגבי יעילות האלגוריתם נגע בזמן הריצה של האלגוריתם. כלומר, כדי לקבוע כמה אלגוריתם מסויים הוא יעיל, חישבנו מה סדר גודל זמן הריצה שלו.

אולם כידוע, למחשב ישנם שני משאבים חשובים – אחד הוא הזמן, ואילו השני הוא הזיכרון. כאשר אלגוריתם רץ הוא צורך זיכרון מהמחשב, ולכן נוכל לשאול את שאלת היעילות בצורה אלטרנטיבית – לאו דווקא כמה זמן האלגוריתם רץ, אלא כמה זיכרון (מקום) הוא צורך בזמן ריצתו.

המדד של כמות הזיכרון הנצרך נקרא **סיבוכיות מקום**. הגדרה יותר מפורטת של סיבוכיות המקום היא: סדר הגודל של מספר תאי הזיכרון שמנצל האלגוריתם בזמן ריצתו.

דוגמא: בהינתן קטע הקוד הבא:

```
int i, n;
for(i=0; i<n; i++)
    System.out.print("*");
```

אנו כבר יודעים שהלולאה הנ"ל רצה n פעמים, ולכן סדר גודל זמן הריצה הוא $O(n)$. מהי סיבוכיות המקום? נספור את תאי הזיכרון שמנצל האלגוריתם: האלגוריתם מגדיר שני תאי זיכרון מגודל `int`. אם רוצים אפשר להתייחס גם למרחב הזיכרון שמוקצה לשיטה עצמה במחשנית. ככה או ככה, מספר תאי הזיכרון הוא קבוע. כלומר, נאמר שסיבוכיות המקום היא $O(1)$ (קבועה).

שימו לב שאנחנו משתמשים באותם סימונים של O גדולה עבור הערכת הסיבוכיות – זמן או מקום.

לעומת זאת, בקטע הקוד הבא:

```
int n;
int[] a = new int[n];
...
```

אנו רואים שהאלגוריתם מגדיר מערך בגודל n . מה סיבוכיות המקום? תלוי ב- n . אם n יהיה 10, אז האלגוריתם יגדיר עשרה תאי זיכרון, ואם n יהיה שווה 100, אז האלגוריתם יגדיר 100 תאים. ככה שהמקום שהאלגוריתם תופס תלוי באופן ישיר בקלט, ולכן נאמר שסיבוכיות הזיכרון היא $O(n)$ (ליניארית).

שאלה – מה תהיה סיבוכיות המקום של אלגוריתם שמגדיר מערך דו-מימדי מגודל $n \times n$?

שימו לב לעניין חשוב בכל הנוגע לסיבוכיות מקום. למשל, בהינתן השיטה הבאה:

```
public void f(int[] a)
{
    int n = a.length;
    System.out.println(n);
}
```

מה סיבוכיות המקום של f ? אמנם השיטה מקבלת כפרמטר מערך בגודל n , אבל חשוב להבחין שהשיטה לא מגדירה את המערך בעצמה ולפיכך אינה אחראית לזיכרון שנצרך כתוצאה מהגדרת מערך זה! כלומר, סיבוכיות המקום היא $O(1)$ (השיטה תופסת מקום קבוע) ולא $O(n)$.

כדי להבין את העניין בצורה יותר אינטואיטיבית ניתן לחשוב שוב על החיפוש הבינארי – אנו יודעים שסדר גודל זמן הריצה של חיפוש בינארי הוא $O(\log n)$. אבל אנחנו גם יודעים שכדי שחיפוש זה יעבוד נכון המערך צריך להיות מלא במספרים, וממובין, אחרת אין שום משמעות לחיפוש הבינארי וממילא הוא לא יעבוד נכון.

כמה זמן לוקח לקלוט n איברים למערך? $O(n)$.

כמה זמן לוקח למיין מערך בגודל n ? $O(n \log n)$.

למה בכל זאת, למרות חסמי הזמן הגדולים הללו, אנחנו עדיין טוענים שזמן החיפוש הבינארי הוא $O(\log n)$? מפני שכל מה שקדם לזמן הפעלת האלגוריתם הוא לא חלק מפעולת האלגוריתם ולכן אינו נכנס לחישובי הזמנים שלו! האלגוריתם של חיפוש בינארי אמור לקבל כקלט מערך מלא במספרים וממובין, אבל זאת לא אחריותו לבצע את קליטת המספרים ומיונם, ולכן פעולות אלו אינן חלק מהזמן שמקדיש האלגוריתם לפתרון הבעיה.

באותה מידה אנו לא נספור את תאי הזיכרון שהוגדרו לפני תחילת האלגוריתם במניין תאי הזיכרון שהאלגוריתם מנצל בזמן פעולתו, ולכן אלגוריתם שמקבל מערך בגודל n כפרמטר לא מגדיר אותו בעצמו, והזיכרון שצורך אותו מערך אינו נחשב לזיכרון שמנצל האלגוריתם. אלגוריתם החיפוש הבינארי, אם כן, צורך תאי זיכרון בודדים וקבועים (לשמירה על האינדקסים של low, high, mid), ולכן סיבוכיות המקום שלו היא $O(1)$.

החישובים למדידת סיבוכיות הזמן דומים מאוד לאלו של סיבוכיות המקום, ולכן אנו לא מרחיבים עליהם שוב, אלא רק נוגעים בהם בקצרה. מכל מקום, מדד סיבוכיות המקום חשוב מאוד בהערכת טיבו של אלגוריתם.