מבני נתונים - 67109
תרגיל 10

להגשה: עד 05.6.19, 23:55

שאלה 1 - גיבוב

נניח שקיבלתם מפתח k לחיפוש בטבלת גיבוב מגודל $m = 2^p$, ויש ברשותכם פונקציית האש h הממפה מפתחות אל המרחב $\{0, 1, ..., m-1\}$. נגדיר את החיפוש כך:

1. נחשב את הערך $j = h(k)$, ונגדיר $i = 0$

   (א) נבדוק את התא ה־j. אם המפתח k נמצא שם נחזיר **אמת**, אם התא ריק נחזיר **שקר** והחיפוש ייגמר.

   (ב) נגדיר $i = i + 1$. אם $i = m$, הטבלה מלאה ונסיים את החיפוש. **אחרת** נגדיר $j = (j + i) \mod m$, ונחזור לשלב 2.

א. הוכיחו כי האלגוריתם הנ"ל הוא למעשה חיפוש ריבועי - Quadratic Probing, ומצאו את הקבועים c_1, c_2 המתאימים לחיפוש.

ב. הראו כי האלגוריתם עובר על כל התאים בטבלה במקרה הכי גרוע.

(**רמז:** הראו כי בכל אחת מה־m איטרציות, הפונקציית האש מחזירה ערך שונה. היעזרו בכך ש־$m = 2^p$, שימו לזוגיות/אי זוגיות של הביטויים שתקבלו, והיעזרו בחשבון מודולו).

שאלה 2 - גיבוב מושלם

בהרצאה ובתרגול ראיתם "גיבוב מושלם במקום ריבועי" - אלגוריתם לגיבוב שבהנתן n מפתחות, יוצר את טבלת הגיבוב וממפה את המפתחות אליה ב־$O(n)$ בתוחלת (כי ראיתם שמספר הפעמים שנצטרך להגריל מחדש את פונקציית הגיבוב ולגבב את המפתחות מחדש קבוע בתוחלת). המקום בזיכרון שהטבלה לוקחת הוא ריבועי ב־n, ולאחר היצירה של הטבלה כל חיפוש מפתח לוקח $O(1)$ (כי אין התנגשויות בכלל).

נרצה לעבוד תחת הנחה אחרת - שבתחילת הריצה אנחנו מקבלים את **מספר** המפתחות שנצטרך להוסיף למבנה הנתונים במהלך הריצה כולה, אבל את המפתחות עצמן נקבל במהלך הריצה באופן מפוזר ואין לנו ידע אילו מפתחות אלה יהיו (רק שהם מתוך מרחב המפתחות שלנו U). נרצה זמן ריצה $O(1)$ לכל הכנסה וחיפוש במהלך הריצה (באופן דטרמיניסטי, כלומר בלי תוחלת הפעם). לטובת זה נאפשר לעצמנו להשתמש במעט יותר זיכרון, כל עוד הוא ליניארי. הבעיה היא שזה לא נראה שנוכל להבטיח את שני הדברים האלה יחד.

1. הציעו מבנה נתונים (ואלגוריתמים) שבהנתן n מספר המפתחות שייכנסו במהלך חייו, מאותחל בזמן ריצה שהוא $O(1)$, ולאחר מכן מאפשר הכנסה וחיפוש מפתחות, כך שבהסתברות הגדולה מ־$\frac{1}{2}$ זמן הריצה של כל אחת מהפעולות הללו היא $O(1)$ לכל מפתח במהלך הריצה.

2. הציעו מבנה נתונים (ואלגוריתמים) באופן דומה לסעיף 1, כך שההסתברות שהכנסה וחיפוש של כל מפתח לאחר האתחול תהיינה בזמן ריצה $O(1)$ תהיה גדולה מ־$\frac{3}{4}$.

3. בהנתן $1 \leq c \in \mathbb{N}$, הציעו מבנה נתונים (ואלגוריתמים) באופן דומה לסעיפים הקודמים, כך שההסתברות שהכנסה וחיפוש של כל מפתח לאחר האתחול תהיינה בזמן ריצה $O(1)$ תהיה גדולה מ־$1 - \frac{1}{2^c}$ כך שמבנה הנתונים יתפוס זיכרון גדול ריבועי ב־n. (רמז: הזיכרון הריבועי שניתן להשתמש בו הוא כפל של n^2 בקבוע שתלוי ב־c. אין צורך למצוא את הקבוע הזה)

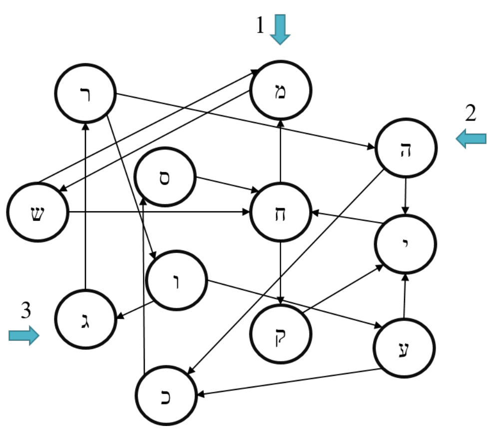
שאלה 3 - שימוש ב־DFS

ביבי רוצה להקים ממשלה. בכנסת יש מספר מסויים של מפלגות, כאשר יש זוגות של מפלגות שלא מוכנות לשבת יחד בממשלה וגם לא באופוזיציה. האם הוא יכול להקים ממשלה? (שימו לב שאין עניין של מושבים וספירתם בהפשטה שלנו, ושאצלנו, לעומת הכנסת, 2 מפלגות שלא יכולות לשבת יחד בקואליציה גם לא יכולות לשבת יחד באופוזיציה. השאלה היא רק אם אפשר להפריד אותן לקואליציה ולאופוזיציה כך שבאף אחת מהן לא יושבות 2 מפלגות שלא מוכנות לשבת יחד).

הציעו אלגוריתם, שבהנתן המפלגות והזוגות שלא יכולות לשבת יחד, מחזיר תשובה - האם ממשלה יכולה לקום או לא? (כלומר האם אפשר לחלק את המפלגות לקואליציה ואופוזיציה כך שכל מפלגה באחת מהן בדיוק ואין זוג מפלגות שלא מוכנות לשבת יחד שנמצאות בקואליציה או באופוזיציה). כתבו את רעיון האלגוריתם ופסאודו־קוד. הסבירו למה הוא עובד (אין צורך בהוכחה ריגורוזית).

שאלה 4 - DFS שאינו רקורסיבי

1. הריצו את האלגוריתם DFS על הגרף שבאיור 1, התחילו מהקודקוד המסומן בחץ מספר 1, לאחר מכן 2 וכו'... ובכל קודקוד, במעבר על כל הצלעות ממנו עברו עליהן לפי סדר אלפביתי של הקודקודים השניים. ציירו את הגרף וציינו ליד כל קודקוד את זמן הכניסה וזמן היציאה של הריצה, וסווגו כל צלע (כתבו B ליד צלע־אחורה, F ליד צלע־קדימה, T ליד צלע־עץ והדגישו אותה ו־C ליד צלע־חוצה). כך תבנו את יער ה־DFS עבור הריצה הזו.



איור 1:

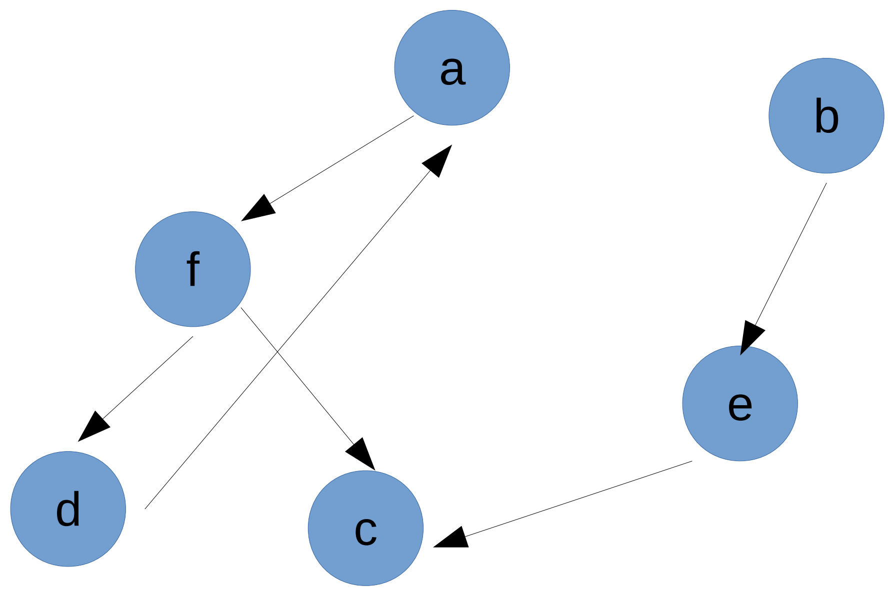
2. מהו זמן הריצה של האלגוריתם? הוכיחו את טענתכם.

3. בכיתה ראיתם מימוש רקורסיבי של האלגוריתם, הסתכלו על הריצה שתיארתם בסעיף הראשון וכתבו פסאודו קוד איטרטיבי של DFS (בגרסא הראשונה שראינו של DFS, זו שאינה מייצרת חותמות־זמן), השתמשו בתשובתכם במחסנית, (מחסנית היא מבנה נתונים המממש מדיניות של $FILO$, יש לה שתי פעולות, $push(x)$ מכניסה את האיבר לראש המחסנית, ו pop מחיקת האיבר הראשון במחסנית והחזרתו)
מהו זמן הריצה של האלגוריתם שהצעתם?

4. החליפו את המחסנית בפתרונכם בתור (תור הוא מבנה נתונים המממש מדיניות $FIFO$, יש לו שתי פעולות, $enqueue$ הכנסת איבר לסוף התור, ו$dequeue$ מחיקת האיבר הראשון בתור והחזרתו), והריצו את האלגוריתם החדש על הגרף מהסעיף הראשון (רק מחץ 2, והפעם אין צורך לסווג צלעות - הסיווג הוגדר לפי אלגוריתם DFS וזה כבר לא אותו אלגוריתם), כעת הריצו את האלגוריתם DFS על הגרף החל מחץ 2 (אין צורך לכתוב את הריצה רק להבין האם ומה ההבדל) האם האלגוריתמים מבצעים את אותה המשימה? אילו הבדלים בהילוך על הגרף אתם מזהים? הסבירו את התופעה. האם זמן הריצה השתנה?

שאלה 5 - חותמות זמן ב-DFS

עבור הגרף שבאיור 2, לכל רשימת חותמות זמן כתבו האם היא יכולה להתקבל מריצת DFS על הגרף או לא:



איור 2: גרף מכוון עליו מריצים DFS בשאלה 5. לאותיות אין משמעות מעבר להיותן שמות לקודקודים.

1.

	a	b	c	d	e	f
pre	8	2	1	7	3	6
$post$	9	5	12	10	4	11

2.

	a	b	c	d	e	f
pre	1	9	7	5	11	3
$post$	2	10	8	6	12	4

.3

	a	b	c	d	e	f
pre	**9**	**1**	**3**	**8**	**2**	**7**
post	**10**	**6**	**4**	**11**	**5**	**12**

.4

	a	b	c	d	e	f
pre	**4**	**11**	**1**	**3**	**9**	**5**
post	**7**	**12**	**2**	**8**	**10**	**6**

.5

	a	b	c	d	e	f
pre	**1**	**9**	**3**	**4**	**10**	**2**
post	**8**	**12**	**6**	**5**	**11**	**7**

.6

	a	b	c	d	e	f
pre	**1**	**11**	**3**	**4**	**9**	**2**
post	**8**	**12**	**5**	**6**	**10**	**7**