

## פרוייקט 2- Hash Tables

עמית אליאסי 316291434

שחר חסקור 208127787

סמסטר אביב 2020

### חלק ב

3.

$$|Q_1| = 3286, |Q_2| = 6571 \quad (\text{א})$$

(ב) בהרצה על QPHashTable נזרקו סך הכל 80 חריגים מסוג "טבלה מלאה", הסיבה לכך היא שנוצרו התנגשויות בסדרת הבדיקות. למעשה הטבלה לא הייתה מלאה, אך השיטה הריבועית אינה עוברת על כל הטבלה בסדרת הבדיקות, אלא על בערך חצי ממנה עבור כל מפתח (כפי שניתן לראות בסעיף א). כך קיים סיכוי ליפול על מפתח עבורו כל התאים בסדרת הבדיקות שלו מלאים, ולמרות שיש מקום בטבלה עדיין לא נוכל למצוא לו מקום. בהרצה על AQPHashTable לא נזרקו חריגים כלל, וזאת, כפי שניתן לראות בסעיף א, כיוון שהפונקציה עוברת על כל המספרים האפשריים בטווח הנתון, כלומר כל עוד יש מקום בטבלה, פונקציית ההאש תגיע מתישהו לתא הריק במהלך סדרת הבדיקות.

(ג) עבור כל  $y \in \mathbb{N}$ ,  $2 < p \in \mathbb{N}_{\text{prime}}$ ,  $y$  נקרא שארית ריבועית של  $p$  אם קיים  $x \in \mathbb{N}$  כך שמתקיים  $x^2 \equiv y \pmod{p}$ . מספר השאריות הריבועיות של  $P$  אם כן, הוא בדיוק גודל הקבוצה  $Q_1$ . ממשפט השאריות הסיני נובע שגודל זה הוא בדיוק  $\frac{p-1}{2}$ , כפי שקיבלנו בסעיף א.

4.

(א)

Class	Running Time (seconds)	Speed Rank
LPHashTable	2.325	3
QPHashTable	2.014	2
AQPHashTable	2.957	4
DoubleHashTable	1.481	1

נשים לב ששיטת "דאבל האשינג" היא המהירה ביותר, וזאת כיוון שהסיכוי להתנגשויות בשיטה זו הוא הכי קטן. כפי שנלמד בכיתה- סדרת הבדיקה לא תלויה רק במיקום הראשוני (התא הראשון אליו הפונקציה שולחת, ב*i*=0), אלא גם במפתח, מה שמאפשר מספר נמוך יותר של בדיקות פרובינג.

אחריה ניתן לראות ששיטת הבדיקה הריבועית היא המהירה ביותר. היתרון בשיטה זו ביחס לשיטה הלינארית הוא שלא נוצרים רצפים, עליהם צריך לעבור עד למציאת תא להכנסה. אך החסרון ביחס

לשיטת ה"דאבל האשינג" הוא שסדרת הבדיקה ל2 מפתחות עם מיקום ראשוני זהה, תהיה זהה (בעיית ההצטברות המשנית).

הבאה בתור, אך לא רחוקה, היא השיטה הלינארית. שיטה זו יוצרת רצפים עליהם נצטרך לעבור בבדי למצוא את התא הריק הבא, מה שמאט את תהליך ההכנסה.

השיטה האיטית ביותר היא האלטרנטיבית, מאותן סיבות של הלינארית והריבועית, וגם בשיטה זו חישוב פונקציית ההאש דורש יותר זמן.

(ב)

Class	Running Time (seconds)	Speed Rank
LPHashTable	4.204	2
AQPHashTable	5.260	3
DoubleHashTable	3.581	1

בסעיף זה לא הרצנו את הניסוי עבור QPHashTable כיוון שעלול להזרק TableIsFullException. הסיבה לכך היא שהפונקציה לא תמיד עוברת על כל התאים, כלומר לפעמים תפספס תאים ריקים (בהם ניתן לשים את האיבר החדש) ו"תחשוב" שאין מקום.

ההבדלים בין הסעיפים הם הגדלת הזמן עבור כל טבלה פי בערך 2, וזאת כיוון שאנחנו מכניסים כמעט פי 2 יותר מפתחות, וכך לא רק מגדילים את מספר ההכנסות, אלא גם מגדילים את הסיכוי ליותר התנגשויות.

נשים לב שזמן הריצה של "דאבל האשינג" גדל בערך פי 2.4, בעוד שזמן הריצה של הטבלאות האלטרנטיביות והלינאריות גדל בערך פי 1.8. הסיבה להבדל זה היא שהיתרון של הדאבל האשינג היה שעבור 2 איברים שונים, הסיכוי לסדרת בדיקות שווה שואף ל0 (בניגוד לשתי השיטות האחרות בהן שני איברים עם מיקום ראשוני זהה יקבלו סדרת בדיקות זהה), כלומר, ההסתברות להתנגשות תלוי במספר המפתחות המוכנסים: ככל שנגדיל את מספר המפתחות כך נקבל יותר ויותר התנגשויות בבדיקות הפרובינג.

5.

Iterations	Running Time (seconds)
First 3 iterations	8.275
Last 3 iterations	18.214

ניתן לראות ש 3 האיטרציות הראשונות התבצעו יותר מהר מ 3 האחרונות. הסיבה לכך היא שככל שיהיו יותר תאים "מחוקים" כך יקח יותר זמן למחוק ולהכניס מפתחות- במעבר על הטבלה יהיו פחות תאים NULL, כלומר נאלץ לעבור על יותר תאים לפני שנוכל לקבוע שהמפתח לא נמצא בטבלה.

לדוגמא: עבור פונקציית ההאש  $f(x, i) = (x \% m + i((x + 1) \% m)) \% m$ , עבור  $m = 10$  ועבור הטבלאות:

(1)

מחוק	1	32	מחוק	מחוק	מחוק	מחוק	מחוק	מחוק	9
------	---	----	------	------	------	------	------	------	---

(2)

	1	32							9
--	---	----	--	--	--	--	--	--	---

נסיון מחיקת האיבר 52 מהטבלה הראשונה יגרור מעבר על כל  $i \in [m]$  בעוד שאותו ניסיון מחיקה בטבלה 2 יסתיים לאחר 2 איטרציות, כשנגיע למקום  $f(52,1) = 7$  נמצא נאל ונבין ש2 לא בטבלה.

בנוסף, לפני כל הכנסה עלינו לחפש תחילה את המפתח ולוודא שאינו בטבלה. חיפוש זה ייסתיים כשנגיע לתא ריק (נאל), כשנמצא את המפתח, או כאשר נעבור על כל האפשרויות בהם המפתח יכול היה להמצא. כלומר במקרה בו כל התאים מחוקים, הסיכוי לסיים את החיפוש במציאת נאל הוא 0, ואם המפתח לא נמצא נאלץ לסיים רק לאחר מעבר על כל האפשרויות.