מבני נתונים - תרגול 11
פונקציית ערבול מושלמת (Perfect Hash)

גלעד אשרוב

9 ביוני 2012

תקציר

בתרגול זה נלמד על $perfect\ hash$ (פונקציית ערבוב מושלמת). המטרה היא בהנתן קבוצה של מפתחות S מתוך עולם U למצוא פונקצית ערבוב **מושלמת**, כלומר, פונקציה ללא התנגשויות כלל בקבוצה הנתונה S. בצורה זו ברור כי החיפוש אורך $O(1)$ זמן. נראה שלם כך כמות הזיכרון נדרש היא בגודל $O(n^2)$, כאשר n הוא מספר האיברים ב - S. בנוסף, נראה שניתן לקבל מבנה לינארי בגודל של S, המשתמש בלכל היותר שתי פונקציות ערבוב לשם מציאת איבר (כלומר, חיפוש עדיין לוקח $O(1)$). נעיר שמבנה הנתונים שלנו הוא סטאטי - אנחנו לא דנים בהוצאה ובהכנסה במסגרת דיוננו זה.

1 הקדמה

נתבונן בבעיה הבאה:
נתונה קבוצה של מספרים $U = \{0, \ldots, u-1\}$, ותת קבוצה $S = \{x_1, \ldots, x_n\} \subseteq U$. נרצה לענות על השאילתות הבאות:

- $query(x)$ - האם $x \in S$. אם יש מידע - החזר (כלומר, x הוא רק מפתח).
- $insert(x)$ - הכנס את x ל- S.
- $delete(x)$ - מחק את x מ- S.

פתרון. פתרון אפשרי לבעיה הוא:

- מערך בגודל U. מעין $bit - vector$. הבעיה - אם $|S| << |U|$, יש לנו המון בזבוז של מקום...
- פונקציית $hash$.

בבואינו לעסוק כעת בפונקציית $hash$, נתעלם משתי הפעולות האחרונות (הכנסה והוצאה), ונדבר רק על הפעולה $query$. למעשה, נדבר על מבנה נתונים **סטטי**.

כיצד נבחר פונקציית $hash$ טובה? נניח $|S| = n$. אנו רוצים לבחור פונקציה שלוקחת אלמנטים מ - U ומעבירה אותם ל - $\mathbb{Z}_m$ (למערך בגודל m נקבע את גודלו מאוחר יותר. נרצה שגודלו יהיה לינארי ב- $|S|$). פונקציה טובה כזו היא פונקציה אקראית לחלוטין. כלומר, לכל איבר ב - U, נבחר איבר באקראי מהקבוצה $\{0, \ldots, m-1\}$. עבור שני איברים x, y, קבועים, ההסתברות להתנגשות היא:

$$\Pr_h[h(x) = h(y)] = \frac{1}{m}$$

חישוב זה הוא למעשה אותה השאלה שהייתה לנו בתחילת הסמסטר - בהינתן שני כדורים שנזרקים לתאים - מהי ההסתברות ששני כדורים נתונים ייפלו לאותו התאה? לשם תזכורת, נסביר מדוע חישוב ההסתברות הוא $1/m$. מתבוננים במספר התא ש - x הגיע אליו, ושואלים מהי ההסתברות ש - y יגיע לאותו התא. מכיוון שהתא אליו y מגיע נבחר באקראי מבין כל התאים, נקבל הסתברות של $1/m$.

בכלל, עבור k איברים מתוך S, נניח $s_1, \ldots, s_k$, ההסתברות שכולם ייתנגשו לאותו התא היא:

$$\Pr_h[h(x_1) = \ldots = h(x_k)] = \left(\frac{1}{m}\right)^{k-1}$$

כאשר ההסתברות נלקחת על הבחירה של h. הסבר לחישוב ההסברות: x_1 קובע תא שאליו כולם צריכים ללכת. ההסתברות שכל אחד מ - $x_2, \ldots, x_k$ נפלו בתא ש - x_1 נפל אליו היא $1/m$. מכיוון שכל המאורעות בלתי תלויים, נקבל $(1/m)^{k-1}$.

הבעיה. כזכור, בחרנו פונקצייה אקראית בשביל פונקציית הערבול ($hash$). כדי לייצג את הפונקציה, לכל ערך ב- U נצטרך לשמור לאיזה תא הוא ממופה, או בעצם - ערך ב - $\mathbb{Z}_m$. בכדי לייצג ערך ב - $\mathbb{Z}_m$ דרושים לנו $\log m$ ביטים (למה?), ולכן, כדי לייצג את כל הפונקצייה אנו נדרשים ל - $|U| \log m$ ביטים, שזה המון!

1.1 דרישות

למעשה, נרצה מספר תכונות מפונקציית ה - $hash$:

- **אתחול:** בהינתן קבוצה S נרצה לבנות בצורה מהירה.
- **זמן:** נרצה לענות על שאילתות בצורה מהירה.
- **מקום:** פונקציית ה - $hash$ צריכה להיות בעלת תיאור קומפקטי. כלומר, ייצוג הפונקצייה צריך להיעשות בעזרת מספר מועט של ביטים.

2 פונקציית ערבוב מושלמת - Perfect Hash

המטרה בסופו של דבר היא להגיע לפונקצייה ערבוב מושלמת. כלומר:

הגדרה 1. בהינתן קבוצה $S \subseteq U$, נאמר שהפונקציה - $h : U \to \mathbb{Z}_m$ היא פונקציית ערבוב (גיבוב) מושלמת אם לכל $x \neq y \in S$, מתקיים $h(x) \neq h(y)$.

במילים אחרות, הפונקציה h היא פונקציית ערבוב מושלמת עבור הקבוצה S אם אין בכלל התנגשויות בתוך הקבוצה.

3 פונקציית hash אוניברסלית

נדבר על משפחה של פונקציות, ונבחר פונקציה אחת מתוך המשפחה באקראי. כאשר דיברנו על בחירת פונקציה אקראית, למעשה דיברנו על כלל הפונקציות $U \to \mathbb{Z}_m$, ואמרנו ש - h נבחרת באקראי מתוכם. עבור משפחה זו, ראינו שקיימות שתי תכונות:

- לכל $x, y \in U$, $x \neq y$, מתקיים:

$$\Pr_h[h(x) = h(y)] = \frac{1}{m}$$

- ובאופן כללי, לכל $x_1, \ldots, x_k \in U$, $x_1 \neq \ldots \neq x_k$, מתקיים:

$$\Pr_h[h(x_1) = \ldots = h(x_k)] = \left(\frac{1}{m}\right)^{k-1}$$

אמרנו שמשפחה זו בעייתית מכיוון שאין לפונקציית הגיבוב ייצוג קומפקטי. נתבונן כעת במשפחה חדשה של פונקציות, המתקיימת רק את התכונה הראשונה. לאחר מכן, נראה שתכונה זו מספיקה לנו, וכמו־כן, שקיימת משפחת פונקציות כאלה בעלי ייצוג קומפקטי. פורמלית, נגדיר:

הגדרה 2. משפחת פונקציות $hash$ $\mathcal{H} = \{h_i \mid h_i : U \to \mathbb{Z}_m\}$ תקרא משפחה אוניברסלית אם לכל $x, y \in U$, $x \neq y$ מתקיים:

$$\Pr_{h \in \mathcal{H}}[h(x) = h(y)] \leq \frac{1}{m}$$

כאשר ההסתברות נלקחת על פני בחירת h מתוך $\mathcal{H}$.

למעשה, המשפחה הקודמת שהגדרנו (כלל הפונקציות מ־ U ל־ $\mathbb{Z}_m$) הייתה עמידה בפני התנגשויות לכל תת קבוצה של איברים. אצלינו - אנחנו עמידים בפני התנגשות רק לזוג.

השאלה הראשונה היא האם קיימת משפחה של פונקציה כזאת. השאלה השנייה־ האם ייצוגה קומפקטי, והשאלה השלישית היא.... למה הסתברות נמוכה יחסית לקבלת התנגשות רק לזוג - באמת מספיק לנו??

ראשית, קיימת משפחה כזאת של פונקציות. נתבונן במשפחה הבאה:

$$\mathcal{H}_{p,m} = \{h_{a,b} \mid 1 \leq a \leq p-1, 0 \leq b \leq b-1\}$$

וכאשר:

$$h_{a,b} = ((ax + b) \bmod p) \bmod m$$

כמה פונקציות במשפחה? עבור p מסויים, יש כאן p^2 פונקציות. הטענה אומרת שהמשפחה הנ"ל היא משפחה אוניברסלית. כלומר, אם a ו־ b נבחרים באקראי כפי שצויין, התכונה שביקשנו־ מתקיימת. ההוכחה היא בספר, מבוא לאלגוריתמים, קורמן, **מהדורה שנייה** (איני בטוח שיש תרגום לעברית של המהדורה הזו..).

נשים לב שייצוג הפונקציה הוא קומפקטי־ בכדי לייצג את הפונקצייה דרושים לנו רק a ו־ b, שלשניהם אנו צריכים $\log p$ ביטים.

4 בניית פונקציית Perfect Hash עבור $S \subseteq U$

כעת, בהינתן קבוצה S נרצה לחפש עבור הקבוצה פונקציה מושלמת. לשם כך, נתבונן באלגוריתם הבא. האלגוריתם בוחר באקראי פונקציה מתוך המשפחה, ופשוט בודק האם היא באמת מושלמת. אם הפונקציה מושלמת־ האלגוריתם מפסיק, ומחזיר את הפונקציה. אם היא לא־ הוא חוזר להתחלה ומחפש פונצקיה מחדש (שוב, באקראי). כפי שנראה בניתוח, ההסתברות שהפונקציה תהיה טובה יהיה לנו מספיק גדול. באופן מפורט יותר:

- **קלט:** הקבוצה $S \subseteq U$ (נניח, נתונה לנו רשימה מקושרת של כל האיברים).
- **פלט רצוי:** מערך מגודל m, ופונקציה $h : U \to \mathbb{Z}_m$, כל שלכל זוג $x \neq y \in S$, $h(x) \neq h(y)$.
- **האלגוריתם:**
  1. נקבע משפחה אוניברסילית בצורה שרירותית (התלויה ב־ m).
  2. נבחר $h \in \mathcal{H}$ בצורה אקראית.
  3. ניצור "קבוצות" (רשימות) של $S_i = \{x \in S \mid h(x) = i\}$. (כלומר, לכל אינדקס, בודקים כמה איברים מופו לאותו האינדקס).
  4. אם קיים i כך ש־ $|S_i| > 1$, חזור ל (2). (אם קיימת איזושהי התנגשות־ בחר פונקציה מחדש).

ניתוח מספר ההתנגשויות. אנו מעוניינים לחשב מהי ההסתברות שכאשר בחרנו פונקציית $hash$ מתוך $\mathcal{H}$, נקבל איזושהי התנגשות. נקבל: (הסבר על החישוב מובא לאחר החישוב)

$$\Pr\left[\exists\ collision\right] = \Pr\left[\exists x,y \in S, x \neq y, h(x) = h(y)\right] \leq \sum_{x,y \in S, x \neq y} \Pr\left[h(x) = h(y)\right] \leq \binom{n}{2} \cdot \frac{1}{m} \leq \frac{n^2}{2m}$$

מספר הערות על חישוב זה:

- החישוב מאוד דומה לניתוח מספר התנגשויות לכפי k כדורים שנזרקים ל-n תאים כפי שראינו בתרגול 3.
- בחישוב זה נעזרנו ב־ $union\ bound$. הכלל אומר כי: $\Pr[A \cup B] \leq \Pr[A] + \Pr[B]$.
- באשר לאי השיוויון השלישי ־ אנו בודקים את כל הזוגות x, y האפשריים שיש, ללא חזרות. ישנן $\binom{n}{2}$ זוגות כאלו. עליהן, אנו מחשבים מה ההסתברות שיש התנגשות בין כל זוג וזוג. ההסתברות להתנגשות של כל זוג היא קטנה מ ־ $1/m$, וזה נובע מתכונת האוניברסליות של $\mathcal{H}$.

אם ניקח $m = n^2$, נקבל כי:

$$\Pr\left[\exists\ collision\right] \leq \frac{n^2}{2m} = \frac{n^2}{2n^2} = \frac{1}{2}$$

כלומר בהסתברות קטנה מחצי - תהיה בעיה.

ניתוח זמן ריצה. בכל סיבוב מבצעים עבודה התלויה באורך של $|S| = n$. השאלה היא כמה סיבובים ישנם. מכיוון שאנחנו עובדים באלגוריתם הסתברותי, זמן הריצה של האלגוריתם יכול להמשיך עד אינסוף (אף פעם לא נמצא פונקציה שאין בה התנגשות...), אבל באופן כללי, אנו **מצפים** שמספר הסיבובים יהיה קטן. למעשה, אינטואיטיבית, מכיוון שההסתברות שקיימת התנגשות לפונקציה אקראית קטנה מ ־ $1/2$, אנו מצפים שנצטרך לבדוק 2 פונקציות עד שנקבל פונקציה טובה.

כאמור, כאשר נבחר $m = n^2$, ההסתברות שנצטרך לחזור לשלב (2) היא קטנה מ ־ $1/2$. לכן, נקבל:

$$E(number\ of\ rounds) = \sum_{i=1}^{\infty} i \cdot \Pr[number\ of\ rounds = i] \leq \sum_{i=1}^{\infty} i \cdot \frac{1}{2^i} = 2$$

כלומר, אנו **מצפים** שלאחר שני סיבובים, האלגוריתם יעצור.

לסיכום, קיבלנו פונקציית $hash$ מושלמת. הבנייה (צפוייה) ב - $O(n)$, הפונקציה יעילה, קלה לחישוב, וקומפקטית. הבעיה היחידה - אנו צריכים $O(n^2)$ מקום!

5 FKS - Fridman Komels Szomeredi

המטרה היא לבחור m שיהיה בגודל לינארי ב - n. לשם כך, נבצע "טריק" שנקרא לעיתים $redirection$. לפני שנתחיל, נתבונן רגע במה שקיבלנו למעלה. אנחנו יודעים שעבור קבוצה בגודל k, ניתן למצוא (בקלות) פונקציה מושלמת לתוך טווח בגדול k^2. בנוסף, אנחנו רואים שההסתברות שתהיינה התנגשות כלשהי היא קטנה מ- $\frac{n^2}{2m}$. לכן, אם ניקח את m להיות בגודל n, אנחנו מצפים לקבל התנגשות. אבל - אוקיי, תהיה לנו התנגשות, השאלה המעניינת היא בעצם - כמה התנגשויות אנחנו מצפים לראות? בסעיף זה נראה שכאשר בוחרים m להיות לינארי ב-n יהיו אמנם התנגשויות, אבל מספר ההתנגשויות יהיה קטן (נניח, סה"כ $O(n)$ התנגשויות, כלומר, סה"כ משהו כמו $O(\sqrt{n})$ איברים מתנגשים). עבור האיברים האלו - שמתנגשים - אנחנו נבנה נבחר פונקצית ערבול מושלמת רק בשבילם, אל תוך מבנה נתונים הדורש $O(\sqrt{n}^2)$ מקום, כלומר $O(n)$. בסה"כ נקבל כי כל מבנה הנתונים שלנו דורש $O(n)$ מקום, וחיפוש איבר לוקח עדיין $O(1)$ זמן. נתחיל עם חישוב תוחלת מספר ההתנגשויות.

חישוב תוחלת מספר ההתנגשויות. נעזר בלינאריות התוחלת. נסמן ב - $X_{a,b}$ משתנה מקרי המקבל 1 אם a ו- b מגיעים לאותו התא ("התנגשו"). נקבל:

$$E\left[X_{a,b}=1\right]=\Pr\left[X_{a,b}=1\right]\cdot 1+\Pr\left[X_{a,b}=0\right]\cdot 0=\Pr\left[h(a)=h(b)\right]\leq\frac{1}{m}$$

יהי X משתנה מקרי המציין את מספר ההתנגשויות שיש לנו בסה"כ. כאמור - $X=\sum_{a,b\in S,a\neq b}X_{i,j}$. לפי לינאריות התוחלת:

$$E\left[X\right]=E\left[\sum_{a,b\in S,a\neq b}X_{a,b}=1\right]=\sum_{a,b\in S,a\neq b}E\left[X_{a,b}=1\right]\leq\frac{n}{2}\cdot\frac{1}{m}=\frac{n^2}{2m}$$

כלומר, תוחלת מספר ההתנגשויות היא $n^2/2m$. מה זה אומר? מהי ההסתברות שיהיו לנו לא יותר מ- 100 התנגשויות? מהי ההסתברות שיהיו לנו t התנגשויות? לשם כך נעזר באי-שיוויון מרקוב.

אי שיוויון מרקוב. אי שיוויון מרקוב עוזר לנו להעריך עד כמה משתנה מקרי "קרוב" לתוחלתו. קשה לנו לחשב את ההסתברות שיהיו לנו 100 התנגשויות, או איזשהו t התנגשויות. אי־שיוויון מרקוב נותן לנו דרך לחסום את ההסתברות הזו, כאשר הדבר היחידי שיש לנו ביד זה תוחלת מספר ההתנגשויות.

טענה 3. (אי שיוויון מרקוב) יהי X משתנה מקרי אי שלילי המקבל ערכים בקבוצה $\{x_1,\ldots\}$, ויהי $a>0$. נקבל:

$$\Pr\left[X\geq a\right]\leq\frac{E[X]}{a}$$

הוכחה: לפי הגדרת תוחלת:

$$E[X]=\sum_x\Pr\left[X=x\right]\cdot x=\sum_{x<a}\Pr\left[X=x\right]\cdot x+\sum_{x\geq a}\Pr\left[X=x\right]\cdot x$$

נשים לב שהביטוי $\sum_{x<a}\Pr\left[X=x\right]\cdot x$ הוא חיובי (הסתברות תמיד חיובית, X הוא אי שלילי), ולכן, ניתן לכתוב:

$$E[X]\geq\sum_{x\geq a}\Pr\left[X=x\right]\cdot x\geq\sum_{x\geq a}\Pr\left[X=x\right]\cdot a=\Pr\left[X\geq a\right]\cdot a$$

כאשר האי השיוויון הראשון נכון מכיוון שכל x בתחום שלנו הוא גדול מ- a. האי השיוויון השני נכון לפי הגדרת הסתברות. מכאן, נסיק:

$$\Pr\left[X\geq a\right]\leq\frac{E[X]}{a}$$

■

נחזור לתוחלת מספר ההתנגשויות. נזכור שאנחנו מצפים ל- $\frac{n^2}{2m}$ התנגשויות. אבל, מהי ההסתברות שיהיו לנו יותר מ-n התנגשויות (כלומר, בסביבות $\sqrt{n}$ איברים שמתנגשים?) נקבל:

$$\Pr\left[\#collisions\geq n\right]\leq\frac{E[\#collisions]}{n}\leq\frac{n^2}{2mn}=\frac{1}{2}$$

כאשר ניקח $m=n$. קיבלנו אם כן כי כאשר בוחרים את m להיות לינארי ב- n, אנחנו מקבלים התנגשויות, אבל מספר ההתנגשויות הוא יחסית קטן (סה"כ n תהנגשויות, כלומר, $\sqrt{n}$ איברים שמתנגשים). לכן, האלגוריתם שלנו יעבוד בצורה הבא: הוא יבחר באקראי פונקציית ערבוב. הוא יבדוק אם היא "טובה" - כשהפעם - טובה אומר שאין הרבה התנגשויות (אנחנו מרשים n התנגשויות בסה"כ). אם הפונקציה לא טובה - הוא יגריל באקראי פונקציה חדשה. אם היא טובה - הוא יטפל באיברים המתנגשים.

אלגוריתם FKS. האלגוריתם עובד בצורה הבאה: בוחרים פונקציה באקראי מתוך המשפחה, כאשר בוחר את m להיות לינארי ב-n. מכיוון ש-m לינארי ב- n, אנחנו מצפים שיהיו התנגשויות, ואפילו - הרבה.
נתבונן במימוש הבא:

1. נבחר משפחה אוניברסלית $\mathcal{H}$ בצורה שרירותית (התלויה ב - m, כאשר m יהיה לינארי ב - n).
2. נבחר פונקצייה $h \in \mathcal{H}$ בצורה אקראית.
3. נבדוק כמה התנגשויות יש: ניצור "קבוצות" כך ש - $S_i = \{x \in S \mid h(x) = i\}$.
4. אם מספר ההתנגשויות הכולל $\leq n$, חזור ל - (2).
5. לכל i, אם $|S_i| \leq 1$, לא עושים כלום. אחרת:
   (א) נבחר $\mathcal{H}_{p,|S_i|^2}$ ונבחר מתוכה פונקציה אקראית h_i. נבדוק אם h_i פונקציה מושלמת.
   (ב) אם h_i פונקציה מושלמת, נקח אותה.
   (ג) אחרת - נחזור ל - (5א).

נתבונן על שלב (4) ונשאל כמה פעמים אנו חוזרים לשלב (2). לשם כך, נחשב: תוחלת מספר ההתנגשויות:

$$E(\#\ collisions) = \sum_{x,y \in S, x \neq y} \Pr[h(x) = h(y)] \leq \binom{n}{2} \cdot \frac{1}{m} \leq \frac{n^2}{2m}$$

כעת, אם נבחר $m = n$, ונשתמש באי שיוויון מרקוב ($\Pr[X \geq a] \leq E(x)/a$), נקבל:

$$\Pr\left[\#\ collisions \geq n\right] \leq \frac{E(\#\ collisions)}{n} \leq \frac{n^2}{2mn} = \frac{1}{2}$$

כפי שכבר ראינו, הנ"ל מראה שאנו מצפים שנחזור פעמיים לשלב (2) עד שנמצא פונקציה טובה.
כעת, נרצה לבדוק כמה פעמים חוזרים על שלב (5). נזכור שאנו מגיעים לשלב (5) רק כאשר מספר ההתנגשויות הכולל קטן מ - n. זמן סיבוב 5א הוא $O(|S_i|^2)$ (בתוחלת). אם נחשב סה"כ, כל הסיבוכים הנ"ל:

$$\sum_{i=1}^{n} |S_i|^2 \leq 4 \sum_{i=1}^{n} \binom{|S_i|}{2} = 4 \cdot (\#collisions) \leq 4 \cdot n$$

כאשר השיוויון השלישי נכון מכיוון שמספר ההתנגשויות ב - S_i הוא $\binom{|S_i|}{2}$.
קיבלנו אם כן, מבנה נתונים שזמן הבנייה שלו היא בתוחלת $O(n)$, גישה - $O(1)$. גודל מבני הנתונים:

$$O(\sum_{i=1}^{n} |S_i|^2) = O(\#\ of\ collisions) = O(n)$$