מבני נתונים 2

14 מרץ 2000

<u>זמן משוערך</u>

יהי A אלגוריתם למימוש מבנה נתונים. נרצה לחסום את זמן הריצה הגרוע ביותר של ביצוע m פעולות על מבנה הנתונים. נראה שבמקרים מסוימים ניתן למצוא חסם טוב יותר מאשר mt, כאשר t הוא זמן הריצה הגרוע ביותר לביצוע פעולה בודדת.

<u>הגדרה</u>

תהיינה $f_1,\ldots,f_k$ הפעולות המותרות על מבנה נתונים. <u>הזמן משוערך לביצוע m פעולות</u> (Amortized Time) הוא הזמן הדרוש לביצוע סדרה כלשהי $op_1,\ldots,op_m$ של m פעולות, כאשר $op_i \in \{f_1,\ldots,f_k\}$.

הערה זמן משוערך לביצוע סדרת פעולות מגדיר את הזמן הממוצע לביצוע פעולה בודדת. יש לשים לב לכך שזהו הזמן הממוצע <u>הגרוע ביותר</u> לפעולה, בכל סדרת פעולות אפשרית, ולא הסיבוכיות הממוצעת של האלגוריתם, שמתקבלת מניתוח הסתברותי וחישוב התוחלת של זמן הריצה על פני כל סדרות הקלט האפשריות.

בהמשך נראה מספר שיטות כלליות למציאת סיבוכיות משוערכת של מבני נתונים, אך ראשית נראה שתי דוגמאות פשוטות הניתנות לניתוח אינטואיטיבי.

<u>דוגמא 1: מחסנית</u>

נתונה מחסנית (LIFO) התומכת בשתי פעולות:
push(x) – הכנסת איבר x לראש המחסנית בזמן $O(1)$.
multipop(k) – הוצאת k איברים מהמחסנית בזמן $O(\min(s,k))$, כאשר s הוא מספר האיברים במחסנית (אם אין k איברים במחסנית, אין מבצעים דבר).

מצב התחלתי: מחסנית ריקה.

שאלה: מהו הזמן הדרוש לביצוע n פעולות על המחסנית?
ניתוח גס: המחסנית מכילה לכל היותר n איברים. לכן, פעולת multipop לוקחת לכל היותר זמן $O(n)$. מכיוון שהסדרה מכילה n פעולות שלכל אחת זמן ביצוע $O(n)$ במקרה הגרוע, זמן הביצוע של הסדרה הוא לכל היותר $O(n^2)$.
ניתוח עדין: כל איבר שהוצא בפעולת multipop הוכנס בעבר בפעולת push. לכן כדי שנוכל לבצע multipop שזמן הביצוע שלו $O(k)$, היינו חייבים לבצע קודם לכן k פעולות push, בזמן $O(1)$ כל אחת. מכאן שבהכרח בוצעו סה"כ $k+1$ פעולות מחסנית בזמן $2k$. לכן זמן הביצוע הממוצע לפעולה הוא $O(1)$ לכל היותר, וסדרה של n פעולות דורשת זמן $O(n)$ לכל היותר.

דוגמא זו מצביעה על מוטיב חוזר בניתוחי זמן משוערך: אף על פי שפעולה בודדת עלולה להיות יקרה, מוכיחים שפעולות יקרות לא יכולות להיות תכופות. כלומר, כדי שתתכן פעולה יקרה, חייבות להתבצע הרבה פעולות זולות, ולכן ניתן לחלק את זמן הביצוע של הפעולות היקרות בין הפעולות הזולות שאפשרו אותן.

דוגמא 2: מונה בינארי

נתון רגיסטר בינרי בן n ביטים:
inc – הגדלת הערך במונה ב- 1

Register → 0110110111
+
1
0110111000

ניתן לראות בבירור שהזמן הדרוש לחיבור 1 שווה למס' ה"אחדים" הרצופים בצד ימין + 1, לכן עבור n ביטים פעולת inc בודדה עלולה לקחת זמן n. האם רצף של m פעולות inc לוקח זמן $O(mn)$ במקרה הגרוע ביותר?

מצב התחלתי: רגיסטר מאופס (כולו '0').
ההסתברות שיש k 1 -ים רצופים בקצה הימני היא $P(t=k)=2^{-k}$.

לכן התוחלת של זמן הביצוע של פעולת inc בודדת היא
$$E(Time(inc))=\sum_i P(t_i=k)\cdot k=\sum_i 2^{-k}k\le 2=O(1)$$.

כעת נראה מספר שיטות פורמליות המסייעות ב"פיזור" הסיבוכיות של הפעולות היקרות בין הפעולות הזולות.

שיטת הפוטנציאל (Tarjan)

שיטת הפוטנציאל היא שיטה כללית למציאת זמן משוערך לסדרת פעולות על מבנה נתונים במקרים בהם לא קל לבצע ניתוח אינטואיטיבי כמו בדוגמאות של מחסנית ומונה בינארי. השיטה מתבססת על מציאת פונקציה ממצבים אפשריים של מבנה הנתונים למספרים אי שליליים:
$\varphi : S \to Z^+$.

נתון:
סדרת פעולות $op_1,\ldots,op_n$
יהי t_i הזמן הדרוש לביצוע הפעולה ה-i (Actual Time).
נסמן ב- S_i את מצב מבנה הנתונים לאחר הפעולה ה-i.
סימונים נוספים:
הפרש הפוטנציאלים לפעולה $\Delta\varphi_i=\varphi(S_i)-\varphi(S_{i-1})$
זמן משוערך לפעולה $a_i=t_i+\Delta\varphi_i$

מכאן שזמן הביצוע של סדרת m הפעולות הוא:
$$TotalTime=\sum_{i=1}^m t_i=\sum_{i=1}^m(a_i-\Delta\varphi_i)=\left(\sum_{i=1}^m a_i\right)-\left(\sum_{i=1}^m(\varphi_i-\varphi_{i-1})\right)=\left(\sum_{i=1}^m a_i\right)+\varphi(s_0)-\varphi(S_m)$$
$$\le\left(\sum_{i=1}^m a_i\right)+\varphi_0$$

האי-שוויון נובע מכך ש- $\varphi(S)\ge 0$ לכל S .

נחזור לדוגמת המחסנית

נגדיר פונקצית פוטנציאל: $\varphi(Stack) = height(Stack)$ (הפוטנציאל של מחסנית שווה למספר האיברים בה).

ובעזרת פונקציה זאת נחשב את הסיבוכיות לסדרת פעולות:
הזמן המשוערך לכל סוג של פעולה הוא

$$a(push) = t(push) + \Delta\varphi(push) = 1 + 1 = 2$$

$$a(multipop) = t(multipop) + \Delta\varphi(multipop) = k - k = 0$$

ולכן חסם על זמן הביצוע הגרוע ביותר של סדרת m פעולות על המחסנית הוא

$$TotalTime \le \left(\sum_{i=1}^{m} a_i \right) + \varphi_0 \le \sum_{i=1}^{m} a(push) = 2m$$

נחזור לדוגמת המונה הבינארי

נגדיר פונקצית פוטנציאל: φ(Register) = מס' האחדים ברגיסטר

יהיה k מספר ה-1-ים הרצופים בקצה הימני של הרגיסטר.
אם $k = 0$, פעולת inc מוסיפה 1 לערך פונקצית הפוטנציאל ולוקחת זמן O(1).
עבור k חיובי, פעולת inc מורידה k מערך הפוטנציאל ולוקחת זמן O(k).

$$t(inc) = k + 1$$

$$\Delta\varphi(inc) = -k$$

$$a(inc) = t(inc) + \Delta\varphi(inc) = k + 1 - (k - 1) = 2$$

$$TotalTime \le \left(\sum_{i=1}^{m} a_i \right) + \varphi_0 \le 2m + n$$

אם התחלנו מרגיסטר מאופס, אז $\varphi_0 = 0$ והזמן הכולל הוא לכל היותר $2m$.

הערה:
יש לחשב את הזמן המשוערך של כל הפעולות. אם היינו מגדירים פונקציית פוטנציאל
φ'(Register) = מס' האחדים הרצופים מימין ברגיסטר

אזי עבור Register = $1100\underbrace{111111}_{k}$

$t(inc) = k + 1$ ו- $\Delta\varphi' = -k$ ולכן $a'(inc) = 1$.

אך לעומת זאת: עבור Register = $1100\underbrace{11111110}_{k}$

$t(inc) = 1$ ו- $\Delta\varphi' = k + 1$ ולכן $a'(inc) = k + 2$.

מכאן אנו רואים שבחירת פונקציית הפוטנציאל φ' אינה מאפשרת להוכיח זמן משוערך $O(1)$.

דוגמא: דפדוף במערכת זכרון היררכית

נתבונן במערכת שבה יש בזכרון מקום ל-k דפים, ושאר הדפים נמצאים על הדיסק. כאשר מתבצעת גישה לדף שאיננו בזכרון, מתבצעת פסיקת דף במהלכה מביאים את הדף הדרוש מהדיסק ושמים אותו בזכרון. אם הזכרון מלא, יש לפנות ממנו את אחד הדפים האחרים. נרצה לקבוע איזה דף יש לפנות כך שמספר פסיקות הדף יהיה מינימלי. בהתחלה הזכרון ריק.

אלגוריתם LRU (Least Recently Used): פנה את הדף שהגישה האחרונה אליו היתה בעבר הרחוק ביותר.

יהי OPT אלגוריתם שיודע מראש את כל סדרת הגישות לדפים ומבצע מדיניות אופטימלית, כלומר מדיניות שמביאה את מספר פסיקות הדף למינימום עבור סדרת גישות זו.

טענה:
מספר פסיקות הדף בביצוע LRU הוא לכל היותר פי k ממספר הפסיקות בביצוע OPT.
הוכחה:
נריץ את שני האלגוריתמים במקביל על שני מחשבים זהים עם אותה סדרת גישות. כאשר תגיע גישה לדף, ניתן ל OPT לשרת אותה קודם (במחשב שלו) ורק אח"כ ניתן ל LRU לטפל בה (במחשב שלו).
נרצה להראות שבכל צעד LRU משלם לכל היותר פי k ממה שמשלם OPT. אם מסתכלים על המחיר האמיתי של גישה בודדת, זה כמובן לא מתקיים מפני שיתכן שאחד האלגוריתמים סופג פסיקת דף והשני לא, ולכן היחס איננו חסום. לכן נשתמש בערך הפחת של המחיר ולא במחיר האמיתי של אותה גישה.

בכדי להגדיר את פונקצית הפוטנציאל, נתבונן במימוש ספציפי של של LRU. האלגוריתם יחזיק את הדפים בזכרון בתור FIFO. כאשר מתבצעת גישה לדף שנמצא בזכרון, הדף יועבר לסוף התור. כאשר מתבצעת גישה לדף שאיננו נמצא בזכרון והזכרון מלא, מפונה הדף שבראש התור והדף החדש מוכנס לסופו.

נגדיר את האינדקס של דף p, $I(p)$: אם p נמצא בזכרון של LRU, האינדקס הוא המיקום של p בתור. אחרת, האינדקס הוא 0. נסמן ב-S את קבוצת הדפים שאינם נמצאים כרגע בזכרון של OPT.
פונקצית הפוטנציאל תהיה: $\varphi = \sum_{p \in S} I(p)$. (אינטואיטיבית, הפונקציה מציינת עד כמה רחוקות קונפיגורציות הזכרון של LRU ו-OPT זו מזו.)

הגישות ל-k הדפים הראשונים אינן גורמות לפסיקות דף, שני האלגוריתמים אינם מפנים דפים ומתנהגים אותו דבר. לכן נתחיל את ניתוח הפחת מהרגע שבו הזכרון של שני האלגוריתמים התמלא. ברגע זה, כל דף שאיננו בזכרון של OPT גם איננו בזכרון של LRU, ולכן $\varphi_0 = 0$. כמו כן, φ פונקציה אי-שלילית ולכן מספיק להראות שבכל צעד, ערך הפחת של המחיר שמשלם LRU הוא לכל היותר k פעמים המחיר (האמיתי) ש-OPT משלם באותו צעד.

גישה לדף p מתבצעת בשני שלבים.
שלב 1: OPT מטפל בגישה, LRU לא עושה כלום.
מכיוון ש LRU לא מבצע שום דבר והאינדקסים של הדפים אינם משתנים, ערך הפחת של המחיר שLRU משלם הוא בעצם שינוי הפוטנציאל הנובע משינוי אפשרי בהרכב הקבוצה S. אם OPT איננו סופג פסיקה, הקבוצה S לא משתנה. מכאן שOPT שילם 0 וגם LRU שילם 0 (בחישוב פחת). אם OPT סופג פסיקה, המחיר שהוא משלם הוא 1. הקבוצה S משתנה: הדף p מוצא ממנה ודף אחר כלשהו, p', נכנס אליה. שינוי הפוטנציאל הנובע מכך הוא $I(p') - I(p) \le I(p') \le k$.

שלב 2: LRU מטפל בגישה, OPT לא עושה כלום.
כעת OPT אינו רץ ולכן משלם 0. נראה שערך הפחת ש-LRU משלם הוא 0. נשים לב שהקבוצה S איננה משתנה.
מקרה 1: p בזכרון של LRU. אז המחיר האמיתי הוא 0. שינוי הפוטנציאל נגרם משינוי האינדקסים של דפים:
א. האינדקס של p עולה ל-k. זה לא משפיע על ערך פונקצית הפוטנציאל מפני שOPT הכניס את p לזכרון בשלב 1, ולכן p איננו ב-S.
ב. האינדקסים של שהיו אחרי p בתור קטנים ב- 1. זה יכול רק להקטין את ערך פונקצית הפוטנציאל.
לכן $\Delta\varphi \le 0$ ומכאן ש- $a \le 0$.

מקרה 2: p איננו בזכרון של LRU. אז המחיר האמיתי הוא 1. השינוי בפוטנציאל הוא:

א. האינדקס של p עולה מ- 0 ל- k. כמו קודם, זה לא משפיע על φ.

ב. האינדקסים של כל הדפים שהיו בזכרון של LRU קטנים ב- 1. לפחות אחד מהם נמצא ב- S מפני ש- p אינו ב- S ואינו בזכרון של LRU: היות שלכל היותר k דפים יכולים להיות מחוץ ל- S בכל רגע נתון, רק $k-1$ מהדפים שבזכרון של LRU יכולים להיות מחוץ ל- S.
לכן $\Delta\varphi \le -1$ ומכאן ש- $a = 1 + \Delta\varphi \le 1 - 1 = 0$.

מסקנה: בכל גישה לזכרון, LRU שילם, בחישוב פחת, לכל היותר פי k ממה ששילם OPT.

<u>דוגמא שלילית</u>
נשתמש בשיטת הפוטנציאל (באופן שגוי) כדי להראות שניתן להכניס איברים לעץ חיפוש בינארי בזמן 0.
בהכנסת איבר מוצאים את העלה המתאים ומחברים את האיבר החדש כבן שלו (ימני או שמאלי - על פי המפתח). לא מבצעים כל פעולות איזון על העץ. נתחיל להכניס איברים לעץ ריק.
עבור צומת v בעץ, נסמן ב- d(v) את המרחק של v מהשורש, כשהמרחק נמדד בצמתים. פונקצית הפוטנציאל תהיה: $\varphi = -\sum_v d(v)$. פעולה המוסיפה את המפתח u לעץ מתבצעת בזמן t=d(u) והשינוי בפוטנציאל הוא -d(u). לכן זמן הפחת הוא a=0. מכאן שהזמן לסדרת הכנסות כלשהי: 0.

איפה טעינו? זמן הפחת של כל פעולה הוא אכן 0, ומתקיים $\varphi_0 = 0$. אבל לא נכון ש- $\varphi_i \ge 0$ ולכן הזמן הכולל איננו חסום על ידי $\sum_i a_i$. במילים אחרות, בחרנו פונקצית פוטנציאל בלתי חוקית.

שיטת האשראי (Credit/Debit)
על פי שיטה זו, לכל איבר נצמיד אשראי – מספר שלם של דולרים. כאשר נבצע פעולה על המבנה שלוקחת זמן t נצטרך לשלם עבורה ב-t דולרים שיילקחו מתוך הדולרים שנמצאים באיברי המבנה. אם במהלך ביצוע הפעולות יש תמיד מספיק אשראי לשלם עבורן, אזי הזמן הכולל לביצוע כל הפעולות $\ge$ מספר הדולרים שהוענקו לכל האיברים.

<u>דוגמא: טבלת Hash</u>
M = גודל טבלת Hash
n = מספר האיברים בטבלא
$\rho = n / M$ (צפיפות הטבלא)
פעולת רה-ארגון (Reorganize) - כאשר צפיפות הטבלא מגיעה ל- $\rho = 80\%$ נכפיל את גודלה.
TotalTime(reorganize)=2M, לכן במקרה הגרוע ביותר הכנסת איבר לוקחת זמן לינארי בגודל הטבלא.

נראה שסדרת הכנסות לוקחת זמן לינארי בגדלה לכל היותר, כלומר הזמן הממוצע להכנסה הוא קבוע.

הנחות:
- כל איבר שמוכנס לטבלא מקבל C$
- פעולת reorganize לטבלא בגודל X עולה X$

במעבר מגודל 2M ל 4M
מאז הרה-ארגון האחרון נכנסו לטבלא ρM איברים חדשים, ולכן יש במערכת לפחות $C\rho M$ דולרים. הארגון יעלה 4M דולר.

כדי שתמיד יהיו מספיק דולרים כדי לבצע רה-ארגון, נרצה שיתקיים
$C\rho M = 4M \Rightarrow C = 4/\rho = \mathrm{O}(1)$

C הוא קבוע ואינו תלוי בגודל הטבלא או במספר האיברים בה.
מסקנה: המחיר של n הכנסות לטבלא כזו הוא $\mathrm{O}(n)$ במקרה הגרוע ביותר.

<u>דוגמא: מונה בינארי</u>
נחזור שוב לדוגמת המונה הבינארי. נדאג שבכל עת יהיה לכל '1' במונה דולר אחד. בפעולת קידום המונה, שינוי ערך בביט אחד עולה דולר אחד. כל '1' שהופך ל- '0' משלם עבור עצמו. בנוסף, יש '0' אחד שהופך ל- '1'. נשלם דולר אחד עבור ההפיכה וניתן דולר אחד ל- '1' החדש. סה"כ, נשלם עבור כל קידום של המונה שני דולר ולכן עלות k קידומים היא 2k.

<u>דוגמא: k מתוך n</u>
נרצה לעבור על כל הצירופים של k מתוך n עצמים נתונים. נייצג כל צירוף כוקטור בינארי באורך n שמכיל k -1 -ים. המטרה היא לעבור על כל הוקטורים האלה.
נגדיר סדר בין הוקטורים: עבור שני וקטורים שונים v_1,v_2, $v_1<v_2$ אם במקום הראשון משמאל בו הם שונים, מופיע '0' ב- v_1 ו- '1' ב-v_2. תרגיל: הראו שזהו סדר טרנזיטיבי מלא.

האלגוריתם:
התחל עם $0^{n-k}1^k$. בכל איטרציה: מצא את רצף ה- '1' הימני ביותר. אם אין '0' לפניו, הגענו לוקטור האחרון, 1^k0^{n-k} . אחרת, עבור מהצירוף $w01^x0^y$ ($x>0$, w רישא כלשהי) לצירוף $w10^{y+1}1^{x-1}$.
תרגיל: הראו שהאלגוריתם עובר על כל הוקטורים לפי הסדר.
תרגיל: הראו באילו מבני נתונים יש להשתמש כדי שבכל איטרציה ניתן יהיה למצוא את הקצה השמאלי של רצף ה- '1'ים הימני ביותר בזמן $\mathrm{O}(1)$.
דוגמת ריצה. עבור n=5, k=3 מיוצרים הוקטורים בסדר הבא, משמאל לימין:
00111, 01011, 01101, 01110, 10011, 10101, 10110, 11001, 11010, 11100

ניתוח סיבוכיות בשיטת האשראי:
נדאג שבכל עת, לכל '1' יהיו דולרים כמספר ה'0'ים שמימינו. בהתחלה אין כסף כי כל ה'1'ים מימין. באיטרציה מסוימת, קיימות שתי אפשרויות:
א. הוקטור הנוכחי הוא מהצורה $w01^x$.
- משלמים דולר כדי למצוא את רצף ה'1'ים הימני ביותר.
- משלמים עוד דולר כדי לעבור ל- $w101^{x-1}$.
- מוסיפים דולר ל'1' שזז שמאלה.

ב. הוקטור הנוכחי הוא מהצורה $w01^x0^y$ כאשר $y>0$.
- משלמים דולר כדי למצוא את רצף ה'1'ים הימני ביותר.
- x ה'1'ים שזזים (כדי לעבור ל- $w10^{y+1}1^{x-1}$) משלמים עבור עצמם.
- מוסיפים דולר ל'1' שזז שמאלה.

סה"כ משלמים בכל איטרציה לכל היותר 3 דולר, ומכאן שזמן הריצה לינארי במספר הצירופים.