שחר פרץ

מבנ"ת 15

4 ביוני 2025

מרצה: טל קליגמן

מיונים מההרצאה

Count Sort

מיון countsort – סופר כמה דברים יש מכל איבר, ואז ממיין לפי זה. כיצד נעשה את זה stable? ניקח את ה־counter של כל איבר, ואז ניקח את הסוכמים החלקיים שלו (כלומר אם המערך היה $[2,1,4]$ כי יש 2 פעמים 0, 1 פעמים 1 ו־4 פעמים 2, ואז הסכומים החלקיים $[2,3,7]$ יהיה האינדקסים של מה יבוא איזה במערך ה־output (נוכל לחשוב על זה כאילו אנחנו מחלקים את ה־output לתיבות מסוגים שונים של איברים). זה stable כי ההכנסה באה לפי הסדר אל המערך המקורי. במעבר השני על המערך, כאשר נרצה למקמם אותם בפלט ולא ב־cunter, נקטין ב־1 את מה שמייצג את האינקס של האובייקט שאנחנו שמים כדי לשדבר החדש שנכניס (אם ניפול על אותו התא) יהיה מקום 1 לפני התא שבא לפנוי.

ההנחה שמאפשרת לעשות את זה, היא שכל המספרים קטנים מ־k כלשהו, והסיבוכיות היא $O(k+n)$.

אין באלגוריתם הזה השוואות בכלל. נשאל אך עצמנו, למה אי אפשר לכתוב משהו במודל ההשוואות דומה לזה שעובד עם מה שראינו? כי דחפנו מספר לטווח בדיד, משהו שבכלל לא קיים במודל ההשוואות. יש דבר בשם bin sort ששומר אוספים (bins) של דברים במקום רק לספור, ואז לאחד את ה־bins. זאת תוך שמירה על סדר פנימי. זה דורש את ההנחה שהאיברים חסומים בקבוע (לדוגמה ציונים שחסומים ב־100).

Radix Sort

מיון לקסיקוגרפי. מתחילים מה־lowest significant digit וממשיכים הלאה. בקיצורים LSD ו־MSD (most significant digit). כאשר n מספרים עם d ספרות חסומים ב־$b-1$. היות המספרים מספרים מאפשר לנו לחסום את הספרות האפשרויות כאשר נמיין על כל ספרה – כי הספרות לא גדולות יותר מ־9 (אין ספרה כזו 10 בבסיס 10). הסיבה שנמיין מ־LSD ל־MSD היא שהמיון האחרון הוא הדבר שלפי נקבע הסדר "הכי חשוב". הוא יוצר סדר לקסיקוגרפי כי המיון הפנימי עם count sort הוא stable.

נבחין שהסיבוכיות היא $O(d(n+b))$. לדוגמה, אם נרצה למיין את כל המספרים $(0,1\dots n^k-1)$ ב־$O(kn)$: אם נעביר את המספרים לבסיס p כלשהו, נקבל שיש $O(\log_p n^k)=O(k\log_p n)$ נקבל סה"כ $O(d(n+b))=O(kn\log_p n)$. בפרט עבור המרה לבסיס $p=n$ (ככה האלגו' יעבוד) נקבל $O(\log_n n^k)=O(k)$ ספרות וסה"כ $O(d(n+b))=O(k(n+n))=O(kn)$. שימו לב שהזחנו כאן את הסיבוכיות של המרת הבסיס לבסיס n.

מיון מהיר

Hoare – 1961.

בקורס הזה, בניגוד למבוא מורחב, לא נחלק לשולש רשימות (קטן, שווה, וגדול) כי זה דופק את הזכרון. בקורס הזה quicksort הוא in-place. עץ הרקורסיה הוא קצת שקול לעץ BST אקראי. נוכל להשתמש ב־BST הזה כדי לנסות להתחיל לנתח את הסיבוכיות.

ננסה להפוך את quicksort ל־in-place. הבעיה: כיצד אנחנו נפתרים מהפילוג ל־3 מערכים. יש שני דרכים לעשות את זה.

Lomuto's Partition

"איזה יפני אחד". בהינתן איזשהו pivot' נחליף אותו לסוף. מכאן ואילך, נעבור לפי הסדר על איברי המערך – אם קיבלנו מספר קטן יותר, נסמן בצבע A, ואם קיבלנו גדול יותר, נסמן בצבע B. בסיטואציה שבה קיבלנו רצף של B ולאחריו, A, נחליף את ה־A הראשון ב־B האחרון. בסוף נחליף את ה־pivot עם ה־B הראשון.

מהנקודה הזו אין בעיה – הקריאה הרקורסיבית ל־quicksort תקבל ℓ,r כאשר ℓ נקודות ההתחלה שלה ו־r נקודת הסיום.

Hoare's Partition

גם כאן נשים את ה־pivot בסוף. קצת חלמתי בהקיץ אז אין לי מושג מה זה עושה. צריך לזכור שהוא משאיר את ה־pivot בסוף ולכן ה־quicksort שמקבל את ה־partition שלו צריך להיות זהיר יותר.

תרגילים

נתון אוסף של n מספרים בין 0 ל־1, כך שאין שני מספרים שהמרחק ביניהם הוא $\frac{1}{n}$ או פחות. מיינו את המספרים בזמן $O(n)$.

פתרון: נבצע ווריאציה של count sort. נחלק את $[0,1]$ לתתי אינטרוואלים בגודל $\frac{1}{n}$. נקבל סה"כ n תתי אינטרוואלים. נשים דברים בתאים "על בסיס הטריק הזה" של להכפיל ב־n ולעשות עיגול למטה.

....

שחר פרץ, 2025