

**Del – פויינטר:**  $O(1)$  W.C מערך:  $O(1)$  ממוצע.  
עבור האש סגור ייקח למלא את הטבלה  $\log B$

## מיונים:

שיטות פשוטות (שטובות למספר קטן מאוד של איברים אפילו יותר מהשיטות המסובכות). לכולן חסם עליון ותחתון של  $n^2$ :

1. Bubble Sort – הצפת הקל כלי מעלה.
2. Insertion Sort – אטרציה ודאגת שכל האיברים 1..n ממוינים (נכניס את i למקום הנכון).
3. Selection Sort – בשלב ב- i בוחרים את הקטן ביותר מבין  $A[i]..A[n]$  ושמים במקום ה- i.

שיטות "חזקות":

**Quick Sort:** בחר פיוט, מצא וארגן נקודה נקודה k כך שהערך הקטנים מופיעים משמאל והערכים הגדולים מופיעים מימינה. מיון את ימין ואת שמאלה רקורסיבית. ביתר פירוט: למציאת פיוט נלקח משמאל ונמצא את הגדול הראשון מבין השניים. לחלוקת הפרטים נשתמש בסמנים R ו- L כך ש- R מתחיל מימין וזו שמאלה ו- L להיפך. נסיע את R שמאלה כל עוד הערכים גדולים שווים מהפיוט. נזיז את L מימנה כל עוד הערכים קטנים מהפיוט. אם  $R=L-1$  נעצור, אחרת נחליף בין האיברים, נזיז את R שמאלה, את L מימנה ונתחיל את הלולאה שוב.

במקרה הגרוע ביותר נבחר כל פעם לפיוט את הגדול ביותר ואז כל פעם המערך יתחלק ל-1 ו-1-n. נקבל  $n^2$ . ישנם לב שפרטישן הוא לינארי. במקרה הממוצע נקבל  $n \log n$ . **(לבדוק אולי כדאי לכתוב את ההוכחה של זה)**. אפשר גם לבצע ב- WC של  $n \log n$  (ראה OS).

**HeapSort:** להכניס לערימה, לעשות ExtractMax n פעמים. ב- WC  $n \log n$ , בממוצע גם אבל קצת יותר גרוע מ- QS. הערה: אם רוצים את k הראשונים  $O(n+k \log n)$ .

**MergeSort:**  $n \log n$ .  
 $MS(A,p,r)$   
 If  $p < r$  then  $\{q \leftarrow \text{floor}((p+r)/2);$   
 $MS(A,p,q); MS(A,q+1,r);$   
 $\text{Merge}(A,p,q,r); \}$

**מיונים שטובים כשידועים משהו על הקלט (במודל ההשוואות):**

**מיון מניה (Count):**  $O(n+k)$  למיון מספרים בתחום 1 עד k. עבור i מ-1 עד k נשומר ב-  $C[i]$  את מספר האברים שקטנים שווים ל- i. נסרוק שוב את A ועבור כל איבר נכניס למערך B בגודל n –  $B[C[A[i]]]$  ונפחית 1 מ-  $C[A[i]]$ .

**מיון בסיס (radix):**  $O(dn)$  ממין n מספרים בני d ספרות כל אחד. נתחילים מהספרה הפחות משמעותית ובכל שלב נושים מיון יציב (למעלה CS). **Bin/ Bucket Sort:**  $O(n)$  מחלקים את הקלט למערך בן n האיברים ל- n קבוצות. נעזרים במערך B בגודל n של n רשימות מקושרות וסורקים את A כדי להכניס כל איבר לרשימה המתאימה. ממנים את הרשימות בעזרת מיון הכנסה.

**חסימי מיון:**

בונים עץ החלטות בו כל צומת מייצג את מרחב האפשרויות. כיוון שהעץ הוא בינארי קיים מסלול שאורכו לפחות  $\log k$ , כאשר k הוא מספר העלים. העלים מייצגים סדרים בסיסיים ואת כולם מספר הסדרים הוא  $k! \geq 2^n$ .  $k \geq \log n$  נוציא לוג ונקבל שאורך המסלול המינימלי גדול מ-  $\log(n!)$ .  $\log(n!) = \Omega(n \log n)$ . אפשר להראות שהחסם התחתון לזמן ממוצע הוא אותו הדבר.

- בכמה זמן ניתן לבנות עץ חיפוש בינארי בעזרת ההשוואות בלבד? חסם עליון: שרשרת ארוכה -  $n \log n$ . חסם תחתון: נניח כי ניתן לעשות זאת ב-  $O(n \log n)$  (קטן) אז נבנה עץ, נבצע inorder וקבלנו מיון ב-  $O(n \log n)$  – סתירה.

- האם ניתן להפוך ערימה לע"ב ב-  $O(n)$  במודל ההשוואות? נניח שכן. נקח n מספרים, נכניס לערימה, זה לוקח  $O(n)$ . נהפוך לע"ב באותה היעילות, נעשה אין-אורדר שוב באותה היעילות וקבלנו מיון ב-  $O(n)$ . קלט: k מערכים ממוינים באורך n. האם אפשר להחזיר מערך גדול ממויין ב-  $O(n \log n)$ ? לא. נניח שכן, נקח n מספרים נהפוך כל אחד למערך, נפעיל את האלג' וקבלנו מיון!

קלט: n מספרים כך שכל מספר בדיוק פעמיים. מטרה: חסם תחתון למיון במודל ההשוואות. מספר העלים בעץ ההחלטות הוא  $n!/2^{n/2}$ . לכן קיים מסלול הגדול מ-  $\log$  של הביטוי האחרון וזה  $n \log n - n/2$  כלומר חסם תחתון  $n \log n$ .

קלט: n מספרים מתוכם  $n/\log n$  שונים. פלט: מיון. אלג': לפי תרגיל מהאש  $O(n)$  בממוצע נקבל את האברים וכמה פעמים הם מופיעים. נמיון את  $n/\log n$  השונים, זה לוקח  $\log(n/\log n)$  וזה  $O(n)$ . נשכפל האברים (עוד  $O(n)$ ).

קלט: n קטעים על קטע סגור. פלט: האם יש זוג שנחתך. אלג': נמיון את כל הזוגות  $(a_i, b_i)$  לפי  $a_1..a_n$ .  $(n \log n)$ , לכל i נבדוק האם  $(a_i, b_i)$  נחתך עם  $(a_{i+1}, b_{i+1})$ . (עוד n).

קלט: כמו קודם. פלט: למצוא נקודה שמוכלת במספר מקסימלי של קטעים. נמיון את  $a_1..a_n$  יחד עם  $b_1..b_n$  ואז לעבוד כמו עם אלג' של פתיחת וסגירת סוגריים.

הצג מ"נ שתומך ב- find, ins, del ב- WC של  $\theta(n \log n)$  וכן בפעולה  $\theta(n)$  – nearest(x,k) – האברים שבערך מוחלט הכי קרובים ל-x. פתרון: עץ 2-3 בעזרת עוקב (logn) נמצא את k האברים הקרובים ביותר. אפשר לייעל אם העלים מוחזקים ברשימה מקושרת  $O(k)$ .

קלט: n מספרים מהתחום  $\{1..n^2\}$ . פלט: מיון. אלג': נשבור כל אחד לשני מספרים באורך n, נבצע על כל החלקים הראשונים ואח"כ על השניים count-sort  $O(n+n) = O(n)$ .

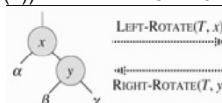
קלט: n מערכים שסך גודלם n, כל המספרים בתוכם 1..k. נרצה למיון כל אחד בפנדר. נאחד את המערכים, נמיון ב- CS ב-  $O(n+k)$  ואז נפצל בחזרה (נשמור זוגות – ערך ושיוך).

**קבוצות עם Merge ו- Find:**

1. נשמור מערך ובו בתא i נשמור את הקבוצה אליה i משתייך.  $\text{find}, O(1)$  – merge,  $O(n)$ .  
 2. נשמור לכל קבוצה את גודלה ואת האיבר הראשון בה ולכל איבר נשמור את הקבוצה אליה הוא משתייך ואת האבא אחריו. merge: עוברים על איברי הקבוצה הקטנה ומשנים את הקבוצה שלהם לגדולה. את האיבר האחרון משרשרים לאיבר הראשון של הגדולה ומשנים את מאפייני הגדולה בהתאם. מאחר וכל איבר שעובר לבעלים חדשים מגדיל אותו לפחות פי 2, אז כל איבר עובר לכל היותר  $\log n$  פעמים במהלך n פעולות merge ולכן הסיבוכיות ל- n פעולות כאלו היא  $n \log n$ .

3. נשמור עץ עבור כל קבוצה בו כל צומת מצביע לאביו (שהוא האיבר הבא בקבוצה). Merge: תולים שורש של אחת הקבוצות על השניה  $O(1)$ . Find: רצים מעלה  $O(n)$ , אך אם נקפיד לתלות עץ קטן על גדול נגדיל את עומק העץ ב-1 בכל תליה ובכל תליה מספר האברים בקבוצה לפחות מוכפל. הסיבוכיות ל- n פעולות  $\text{find}$  היא  $n \log n$ .

**עצים אדומים-שחורים:** עץ בינארי המקיים: כל צומת הוא אדום או שחור; אם צומת הוא אדום, שני בניו שחורים; כל המסלולים הפשוטים מצומת לצאצאים עלים מכילים אותו מספר של צמתים שחורים. כמסקנה מתקבל שזהו עץ בערך מאוזן! צריך לדאוג שחוקיות העץ תמיד נשמרת בפעולות  $\text{ins}, \text{del}$  וזאת מבצעים ע"י רוטציה שמאלית/ימנית  $O(1)$ :



**הכנסה:**  $O(n \log n)$ . תמיד נכניס אדום כאחד העלים לכן רק תכונת אי סימיות האדומים יכולה להיות מופרת. על כן נתקן רקורסיבית כלפי מעלה לפי מקרים. מקרה 1: הדוד של הצומת שהכנסנו גם הוא אדום: צובעים את האבא והדוד בשחור, את הסבא בשחור וממשיכים. מקרה 2: הצומת שהכנסנו הוא בן ימני – נבצע רוטציה שמאלית לאביו. מקרה 3: הצומת שהכנסנו הוא בן שמאל – נבצע רוטציה ימנית לסבו. מחיקה: יש לחקוק כמו בע"ב. אם מחקנו (בפועל) צומת אדום אז אין בעיה אבל אם מחקנו צומת שחור ייתכן כי הפרנו את האיזון ואז נבצע תיקונים  $(\text{del}, \text{fix})$ .

נניח ש-X הוא המקום שמעלינו בוצעה ההשמטה, אחיו (אחרי המחיקה). מקרה 1: W אדום – החלף צבעי W, אבי X, ובצע רוטציה שמאלית לאביו של X. (מוביל לכל אחד מהמקרים). מקרה 2: W שחור 2 בנינים שחורים – הורד שחור מ- W ו- X והוסף לאבי X. מקרה 3: W שחור, בן שמאל אדום ובן ימני שחור – החלף צבעים בין W לבנו השמאלי ובצע רוטציה ימנית (מוליך למצב 4). מקרה 4: W שחור בן ימני אדום. שנה צבעים ובצע רוט' שמאלית לאבי X (סיום הלולאה).

בניית עץ אדום-שחור ממערך: נסדר כמו ערימה, נעבור ב- inorder רק במקום להדפיס נקח את הערך הבא במערך. נצבע הכל בשחור חוץ מהשורה האחרונה אם היא לא מלאה לחלוטין ואז נצבע אותה באדום.

## נוסחאות רקורסיביות נחמדות:

|                                          |                                                               |
|------------------------------------------|---------------------------------------------------------------|
| דורי גודל:                               |                                                               |
| $\log n$                                 | $T(n) = 2T(\frac{n}{2}) + \theta(n) = \theta(n \log n)$       |
| $\log^2 n$                               | $T(n) = 2T(\frac{n}{2}) + \theta(1) = \theta(n)$              |
| $[\log(\log n)]^{\log(\log n)}$          | $T(n) = \sqrt{n} T(\sqrt{n}) + n = \theta(n \lg(\lg n))$      |
| $2^{\log_2 n}$                           | $T(n) = 2T(\frac{n}{2}) + \lg(n) = \theta(\lg(n) \lg(\lg n))$ |
| $(\log n)^{\frac{\log n}{\log(\log n)}}$ | $T(n) = 3T(\frac{n}{4}) + n = \theta(n)$                      |
| $n \log n, \log(n!)$                     | $T(n) = T(n/2) + T(2n/3) + n = \theta(n \lg n)$               |
| $\frac{2n+1}{n-1}, \frac{n^4+1}{n^2+1}$  | $T(n) = T(n/3) - 4 + 1 = \theta(n)$                           |
| $2^{\log^2 n}$                           | $T(n) = T(n-1) + \theta(n) = \theta(n^2)$                     |
| $n^{2 \log n}$                           | $T(n) = T(3n/4) + T(n/4) + \theta(n) = \theta(n \lg n)$       |
| $2^n$                                    | $T(n) = T(9n/10) + T(n/10) + n = \theta(n \lg n)$             |
| $3^n$                                    | $T(n) = T(n/5) + T(7n/10) + 6 + \theta(n) = \theta(n)$        |
| $n!$                                     | $T(n) = T(n/2) + T(n/7) + 6n = \theta(n)$                     |
| $(n+1)!$                                 | $T(n) = T(2n/3) + T(n/3) + n = \theta(n \lg n)$               |
| $n^n$                                    | $T(n) = 2T(\frac{n}{2}) + \sqrt{n} = \theta(n)$               |
|                                          | $T(n) = T(\frac{n}{2}) + T(\frac{n}{7}) + n = \theta(n)$      |
|                                          | $T(n) = T(n-1) + \frac{1}{n} = \theta(\log n)$                |
|                                          | $T(n) = T(\frac{n}{3} - 4) + 1 = \theta(\log n)$              |
|                                          | $T(n) = 9T(n/3) + n = \theta(n^2)$                            |
|                                          | $T(n) = T(2n/3) + 1 = \theta(\lg n)$                          |
|                                          | $T(n) = 3T(n/4) + n \lg n = \theta(n \lg n)$                  |
|                                          | $T(n) = T(n/2) + c = \theta(\lg n)$                           |
|                                          | $T(n) = T(n-1) + \lg n = \theta(n \lg n)$                     |
|                                          | $T(n) = 2T(n/2) + \theta(n^{0.6}) = \theta(n)$                |
|                                          | $T(n) = T(n/3) - 4 + 1 = \theta(\lg n)$                       |

$$1 = O_1(n^{1/\log n}) = O(\log(\log n)) = O(\sqrt{\log n}) = O(\log n) = O(n^{1/3})$$

$$= O(n) = O(2^{\log n}) = O(n^2) = O(n^2) = O(n^2) = O(n^2)$$

$$= O(n) = O(3/2^{\log n}) = O(n^{2/3}) = O(n^{2/3}) = O(n^{2/3}) = O(n^{2/3})$$

$$\log(n!) = \Theta(n \log n) \quad \log n = \Theta(\lg n)$$

$$\sum_{i=1}^n \frac{1}{i} = \Theta(\log(n)) \quad \sum_{i=1}^n \frac{1}{i^2} = \Theta(1)$$

$$\sum_{i=1}^n \frac{1}{i} = \Theta(\log n) \quad \sum_{i=1}^n \frac{1}{i^2} = \Theta(1)$$

- תרגיל לא קשור: מציאת חציון משותף של שני מערכים ממוינים ללא חזרות. אלג': מציאת שני החציונים ובדיקה מי יותר גדול. במערך עם הגדול נפסול את הגדולים ממנו ובמערך עם הקטן נפסול את הקטנים ממנו ונמשיך רקורסיבית.  $O(\log n)$ .