1 – הפרד ומשול

נוסחת האב: ניתוח זמן ריצה של נוסחאות נסיגה מהצורה: $f(n)=a\cdot f\left(\frac{n}{b}\right)+n^c$.

$$f(n)=\begin{cases} c<\log_b a \rightarrow \theta(n^{\log_b a}) \\ c=\log_b a \rightarrow \theta(n^c \log n) \\ c>\log_b a \rightarrow \theta(n^c) \end{cases}$$

אם הפונקציה מצורה אחרת, לפעמים אפשר לעשות הצבה חוזרת.
אם זה לא עובד, יש את שיטת הניחוש: "ננחש" שמתקיים $f(n)\le c\cdot n$ עבור $c\in\mathbb{R}$ כלשהו, ונוכיח באינדוקציה.
(אפשר גם לנחש ש $f(n)\le c\cdot n^2,\ \ f(n)\le c\cdot \log n$ וכו').

לדוגמה: $f(n)=f\left(\frac{3n}{5}\right)+f\left(\frac{n}{3}\right)+7n$. נניח ש $f(k)\le c\cdot k$ עבור $k<n$ כלשהו. אז לדוגמה, $f\left(\frac{3n}{5}\right)\le c\cdot\frac{3n}{5}$

$$f(n)\le c\cdot\frac{3n}{5}+c\cdot\frac{n}{3}+7n=cn\left(\frac{9}{15}+\frac{5}{15}\right)+7n=\frac{14}{15}cn+7n$$

אנחנו רוצים c כך ש: $\frac{14}{15}cn+7n\le cn\rightarrow 7n\le\frac{1}{15}cn\rightarrow 105\le c$

חוקים שימושיים:

מתקיים: $\log_x n=\log_x y\cdot\log_y n$. אז אם x,y קבועים, מעבר של בסיס לוג זה פשוט הכפלה בקבוע. אז הסיבוכיות לא משתנה.

$$x^{\log_y z}=z^{\log_y x},\qquad 1+a+\cdots+a^k=\frac{a^{k+1}-1}{a-1},\qquad 1+a+a^2+\cdots=\frac{1}{1-a}\quad(a<1)$$

כפל בינארי: כדי להכפיל את X, Y, בשיטה ידנית (כפל ארוך) זה $\theta(n^2)$ פעולות על ביטים. אפשר לייעל את זה: נחלק את המספר לX_1,X_2: $\underbrace{\qquad\qquad}_{X_1}\underbrace{\qquad\qquad}_{X_2}$

מתקיים: $X=X_1\cdot 2^{(n/2)}+X_2$. אז $XY=X_2Y_2+2^{\left(\frac{n}{2}\right)}(X_1Y_2+X_2Y_1)+2^nX_1Y_1$.

אם נחשב לפי הנוסחה הזאת, יוצא סיבוכיות לפי נוסחת האב: $T(n)=4T\left(\frac{n}{2}\right)+O(n)=\theta(n^2)$. $a=4,b=2,c=1$.

לא שיפרנו את הסיבוכיות. ננסה שיטה טובה יותר: נגדיר: $A=X_1Y_1,\ \ B=X_2Y_2,\ \ C=(X_1+X_2)(Y_1+Y_2)$ ומתקיים:
$XY=B+n^{\frac{n}{2}}(C-A-B)+2^nA$ כלומר לפי נוסחת האב, $T(n)=3T\left(\frac{n}{2}\right)+O(n)=\theta\left(n^{\log_2 3}\right)<\theta(n^2)$

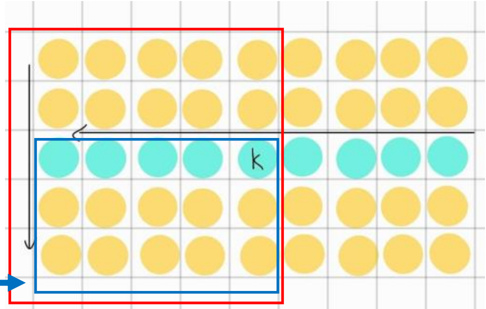
חציון: אם פשוט נמיין וניקח את האיבר האמצעי, זה $O(n\log n)$. ננסה שיטת הפרד ומשול: באופן רקורסיבי, נמצא את החציון של A_1,A_2 , שני החצאים של המערך המקורי. ניקח את B: ה- n/2 איברים האמצעיים, ונמצא את החציון שלו. סיבוכיות: $T(n)=3T\left(\frac{n}{2}\right)+O(n)$, שזה יותר מ$O(n\log n)$**.** ננסה שיטה טובה יותר, שפותרת בעיה כללית יותר: אלגוריתם $\mathrm{select}(A,t)$ מוצא את האיבר ה-k בגודלו במערך:

נקבל את K מהפונקציה pivot . נסדר מחדש את A, לפי כל מה שקטן או גדול מ k. אם יש בדיוק t-1 איברים קטנים מ k**,** סיימנו. אחרת, נקרא רקורסיבית ל select לפי מספר האיברים שקטנים מ k :

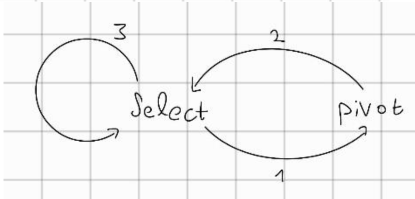
```
Select(A, t)
    K ← pivot(A)
    Re-order A around k: S1 < k < S2. Takes O(n) time.
    if( |S1| == t-1) return k
    if( |S1| > t-1) return select(S1, t)
    if( |S1| < t-1) return select(S2, t - |S1| - 1)
```

נחלק את המערך לקבוצות של 5 ונמיין כל אחת. בגלל שממיינים קבוצה קטנה בגודל קבוע, הסיבוכיות קבועה. נבנה את B, החציונים מתוך כל קבוצה. נחזיר את select על B.

```
pivot(A)
    divide A into groups of 5
    sort each group. Takes O(n) time.
    B ← medians from each group.
    Return Select(B, n/10)
```



ננתח את הסיבוכיות של פיווט: יש n/5 קבוצות. לחצי מהן – n/10 – יש חציון שקטן או שווה ל k. בכל קבוצה כזאת, יש 3 איברים שקטנים או שווים ל k.



בסה"כ, יש לפחות 3n/10 קטנים מ k, וגם 3n/10 גדולים מ k.אז במקרה הכי גרוע, אם תמיד נקבל k הכי קטן שאפשר, בקריאה הרקורסיבית הבאה נקרא לפונקציה על מערך בגודל 7n/10 לכל היותר. אז הנוסחה הרקורסיבית היא:

$$T(n) = \underbrace{O(n) + T(n/5)}_{\text{select}(B)} + T\left(\frac{7n}{10}\right) = O(n)$$

(ההוכחה היא באינדוקציה).

זה גם נותן לנו דרך "לתקן" את הבעיה של quicksort. נשתמש בפונקצית חציון כדי לבחור פיווט טוב בכל שלב.

<u>**קידומת משותפת ארוכה ביותר:**</u> קלט: מערך של n מחרוזות. בשיטת הפרד ומשול, רקורסיבית:

$$T(n) = \begin{cases} O(1), & n \le 2 \\ 2T\left(\frac{n}{2}\right) + O(1), & \theta(n) \end{cases}$$

לפי נוסחת האב.

<u>**תת מערך גדול ביותר:**</u> הקלט: מערך של n מספרים חיוביים ושליליים. פלט: תת מערך רצוף עם הסכום הכי גדול.
נמצא את המערך הגדול בכל חצי, רקורסיבית. נבדוק גם מה הכי גדול: כל צד ימין, כל צד שמאל, או תת-המערך מהאמצע עד נקודה מסוימת בימין או שמאל. ניקח את המקסימלי מבין כל אלה. סה"כ: $T(n) = 2T\left(\frac{n}{2}\right) + O(1) = O(n\log n)$

<u>**ספירת הפיכות סדר במערך:**</u> הפיכת סדר היא זוג כך ש:
$i < j,\ A[i] < A[j]$.
בשיטה הנאיבית – נבדוק את כל הזוגות, $O(n^2)$. ננסה הפרד ומשול.
מתבסס על mergeSort. בזמן הmergen, בכל פעם שניקח איבר מהמערך הימני לפני השמאלי, זה אומר שהייתה הפיכת סדר, אז נוסיף לספירה את מספר האיברים שנשארו במערך השמאלי.