1

$$\overset{\overset{i}{\uparrow}}{}[\overset{\overset{j,p}{\uparrow}}{60}, 70, 80, 90, 100, 1, 5, 9, 11, 15, 19, 21, 25, 29, \overset{\overset{r}{\uparrow}}{30}] \xrightarrow{\text{partition}}$$

$$[\overset{\overset{i}{\uparrow}}{1}, 80, 90, 100, \overset{\overset{j}{\uparrow}}{70}, 5, 9, \ldots] \rightarrow$$

$$[1, 5, 9, \overset{\overset{i}{\uparrow}}{11}, 70, 80, 90, 100, \overset{\overset{j}{\uparrow}}{15}, 19, 21, \ldots] \rightarrow \cdots \rightarrow$$

$$[1, 5, 9, 11, 15, 19, 21, 25, \overset{\overset{i}{\uparrow}}{29}, \overset{\overset{q}{\uparrow}}{30}, 90, 100, 70, \overset{\overset{j}{\uparrow}}{80}] \xrightarrow{\text{quicksort(A,p,q-1)}}$$

$$[\overset{\overset{p}{\uparrow}}{1}, 5, 9, 11, 15, 19, 21, 25, \overset{\overset{r}{\uparrow}}{29}, \ldots] \rightarrow \cdots \rightarrow$$

$$\overset{\overset{r}{\uparrow}}{}[\overset{\overset{p}{\uparrow}}{1}, 5, 9, 11, 15, 19, 21, 25, 29, \ldots] \xrightarrow{\text{quicksort(A,q+1,r)}}$$

$$[\ldots, \overset{\overset{p}{\uparrow}}{90}, 100, 70, \overset{\overset{r}{\uparrow}}{80}] \xrightarrow{\text{partition}} [\ldots, 70, \overset{\overset{q}{\uparrow}}{80}, 90, 100] \rightarrow \cdots \rightarrow$$

$$[1, 5, 9, 11, 15, 19, 21, 25, 29, 30, 60, 70, 80, 90, 100]$$

2 א נוכיח שבהינתן תת מערך $A[p..r]$ ו-$y = A[p]$, קריאה ל-Partition1 תסדר את תת המערך כך $A[\underbrace{\ldots}_{<y}, y, \underbrace{\ldots}_{>y}]$.

נגדיר ללולאה שבשורות 4-12 את שמורת הלולאה הבאה:

שמורת לולאה: בתחילת הלולאה לכל $p + 1 \leq k \leq i - 1$ מתקיים $A[k] < y$ וכן לכל $j + 1 \leq k \leq r$ מתקיים $A[k] > y$.

אתחול: לפני הכניסה ללולאה מתקיים $i = p$ ו-$j = r + 1$ והשמורה מתקיימת באופן ריק.

תחזוקה: הלולאה בשורות 6-7 מזיזה את i קדימה עד שמגיעים לאיבר המקיים $A[i] > y$. בצורה דומה הלולאה בשורות 9-10 מזיזה את j אחורה לאיבר הראשון המקיים $A[j] < y$. התנאי בשורה 11 יתקיים אם מצאנו איבר גדול מ-y ואיבר קטן מ-y כך שהגדול מביניהם נמצא לפני הקטן. במקרה זה, נחליף ביניהם.

סיום: הלולאה מסתיימת כאשר $j \geq i$ אבל מכיוון ש-i, j עוצרים באיבר הראשון שגדול/קטן מ-y ומכך ש-$A[j + 1] > y$ ו-$A[i - 1] < y$ נקבל ש-$i = j + 1$ ולכן לפי שמורת הלולאה נסיק ש-$A[p + 1..j] < y$ וכן $A[j + 1..r] > y$. בשורה 13 יוחלפו

$A[j] \leftrightarrow A[p]$ כך שסה"כ נקבל שתת המערך $A[p..r]$ נראה כך:

$$A[\underbrace{\dots}_{<y},\overset{\overset{\text{j}}{\uparrow}}{y},\underbrace{\dots}_{>y}]$$

נניח שהשגרה $\textsc{Quicksort1}$ ממיינת נכונה מערך שגודלו קטן ממש מ-n. מערך בגודל 1 ממויין טריוויאלית. בהינתן מערך בגודל n, בשורות 3-4 אנחנו ממיינים מערך שגודלו לכל היותר $n-1$ ולכן לפי ההנחה השגרה תמיין נכונה את שני תת-המערכים. בגלל שכל האיברים בתת המערך השמאלי קטנים מכל האיברים בתת-המערך הימני (כי קראנו ל-$\textsc{Partition1}$), המערך כולו ממויין.

2 ב הבדיקה $p \geq j$ מיותרת שכן הלולאה בשורות 9-10 מחסירה מ-j בכל איטרציה 1 ובשלב מסוים יתקיים $j = p$ והתנאי $A[j] > y = A[p]$ לא יתקיים והלולאה תיעצר.

2 ג בכל קריאה ל-$\textsc{Partition1}$ ניכנס ללולאה הראשית פעם אחת, תנאי הלולאה מוסיף 2 השוואות. לא ניכנס ללולאה בשורות 6-7, בדיקת התנאי תוסיף עוד 2 השוואות. הלולאה בשורות 9-10 תבוצע עד ש-$j = p$ כי לכל $p < j$ מתקיים $A[j] > A[p]$, כלומר סה"כ $2(r-p+1)$ השוואות. שורה 11 תוסיף לנו עוד השוואה. סה"כ $\textsc{Partition1}$ עושה $2+2+2n+1 = 5+2n$ השוואות על מערך בגודל $n = r-p+1$. בשגרה $\textsc{Quicksort1}$ תתווסף לנו השוואה אחת כאשר כל פעם המערך יחולק לשני תתי מערכים בגודל 0 ו-$n-1$.

את מספר ההשוואות כפונקציה של גודל המערך נתאר באמצעות הפונקציה:

$$\begin{aligned}
C(n) &= C(n-1) + C(0) + 5 + 2n + 1 \overset{\overset{C(0)=1}{\uparrow}}{=} C(n-1) + 2n + 7 = \\
&= \cdots = C(n-i) + \sum_{k=0}^{i-1} 2(n-k) + 7 = \\
&= C(1) + 7(n-1) + 2\sum_{k=0}^{n-2} n-k = \\
&= 7n - 6 + (n-1)(n+2) = n^2 + 8n - 8
\end{aligned}$$

3 נסביר את דרך הפעולה של האלגוריתם: נסתכל בכל שלב על שני איברים, נניח $A[i], B[j]$. נניח בנוסף ללא הגבלת הכלליות כי $A[i] > B[j]$. על $A[i]$ נוכל לומר שערך המיקום שלו ב-$A \cup B$ הוא **לפחות** $i+j$ (זה נכון מאחר והמערכים ממויינים). בנוגע ל-$B[j]$ נוכל להגיד בצורה דומה שערך המיקום שלו הוא **לכל היותר** $i+j-1$.

אם $k < i+j$, כלומר קטן מערך המיקום המינימלי של האיבר הגדול, האיבר שאנו מחפשים לא נמצא ב-$A[i..n]$. ואם $k \geq i+j$ אז הוא לא נמצא ב-$B[1..j]$. כך בכל שלב של האלגוריתם אנו פוסלים כחצי מאחד המערכים וחוזרים לבעיה המקורית (אם פסלנו חצי תחתון של מערך, עלינו למצוא את ערך המיקום ה-(מספרים האיברים שנפסלו) $- k$).

התהליך יפסק כשאחד המערכים ריק ואז פשוט נחזיר את האיבר ה-k במערך השני.

procedure $\textsc{KthUnion}(A, pa, qa, B, pb, qb, k)$
  if $pa > qa$ **then**
    return $B[pb + k - 1]$

if $pb > qb$ **then**
  return $A[pa + k - 1]$
$mida \leftarrow \lfloor (pa + qa)/2 \rfloor$
$midb \leftarrow \lfloor (pb + qb)/2 \rfloor$
$p \leftarrow mida - pa + midb - pb + 2$
if $A[mida] > B[midb]$ **then**
  if $k < p$ **then**
    return $\textsc{KthUnion}(A, pa, mida - 1, B, pb, qb, k)$
  else
    return $\textsc{KthUnion}(A, pa, qa, B, midb + 1, qb, k - (midb + 1 - pb))$
else
  if $k < p$ **then**
    return $\textsc{KthUnion}(A, pa, qa, B, pb, midb - 1, k)$
  else
    return $\textsc{KthUnion}(A, mida + 1, qa, B, pb, qb, k - (mida + 1 - pa))$

כדי לקבל את הסיבוכיות הנדרשת, נשים לב שהאיבר המבוקש אינו יכול להימצא ב-$A[k+1..n], B[k+1..n]$, ולכן אם $k < n$ ניתן לצמצם את תחום החיפוש ההתחלתי לשני מערכים בגודל k. כמו כן, אם $k > n$ הוא לא ב-$A[1..d], B[1..d]$ כאשר $d = k - n - 1$.

את הקריאה הראשונית נבצע כך:

$d \leftarrow 0$
if $k > n$ **then**
  $d \leftarrow k - n - 1$
$\textsc{KthUnion}(A, 1 + d, \min(k, length[A]), B, 1 + d, \min(k, length[B]), k - 2d)$

בכל קריאה לשגרה תחום החיפוש מצטמצם בחצי אחד המערכים ולכן סה"כ זמן הריצה הוא $\Theta(\lg \min(k, n - (k - n) + 1)) = \Theta(\lg \min(k, 2n - k))$.

4 השגרה שלנו זהה לשגרה $\textsc{Select-Os}$ מעמ' 109 במדריך הלמידה. נראה שגם במקרה ש-$\textsc{Black-Box}$ מחזירה את ערך המיקום ה-$\lfloor \frac{in}{m} \rfloor$ כאשר $1 \le i < m$ סיבוכיות השגרה היא לינארית ב-n.

נוסחת הנסיגה של השגרה היא $T(n) = T(\max(\lfloor \frac{in}{m} \rfloor - 1, n - (\lfloor \frac{in}{m} \rfloor + 1))) + \Theta(n)$. יהי $an \in \Theta(n)$ ונניח שקיים c כך ש-$T(n) \le cn$, אז:

$$\begin{aligned}
T(n) &= T(\max(\left\lfloor \frac{in}{m} \right\rfloor - 1, n - (\left\lfloor \frac{in}{m} \right\rfloor + 1))) + an \\
&\leq c \cdot \max(\left\lfloor \frac{in}{m} \right\rfloor - 1, n - (\left\lfloor \frac{in}{m} \right\rfloor + 1)) + an \\
&\leq c \cdot \max\left(\frac{in}{m}, n - \frac{in}{m}\right) + an \\
&\leq cn \cdot \max\left(\frac{i}{m}, 1 - \frac{i}{m}\right) + an \\
&= cqn + an = cn - (1-q)cn + an = cn + (an - (1-q)cn) \\
&\downarrow \\
\max(\tfrac{i}{m}, 1 - \tfrac{i}{m}) &= q < 1
\end{aligned}$$

אם נבחר $c \geq \frac{a}{1-q}$ אז $an - (1-q)cn \leq 0$ לכל $n > 0$ ולכן $T(n) \leq cn$ וקיבלנו ש-$T(n) = \Theta(n)$ כפי שרצינו.

5 נמצא את החציון לפי x (בפועל זה דורש שינוי קל לשגרות Select ו-Partition). נניח שהוא x_i ונסתכל על המערך לאחר החלוקה. כל הזוגות שנמצאים לפני x_i, ערך ה-x שלהם קטן ממנו, כלומר יש $s = \sum_{j=1}^{i-1} w_j$ איברים קטנים מ-x_i ברשימה, לכן ערך המיקום שלו מתחיל מ-$s+1$ ומסתיים ב-$s+w_i$. אם k נופל בטווח הזה, אז x_i הוא האיבר המבוקש. אם $k \leq s$, אז האיבר המבוקש נמצא בתת המערך השמאלי ל-x_i, ואם $k > s + w_i$ אז האיבר המבוקש נמצא בתת המערך הימני ל-x_i.

1: **procedure** SelectPairs(A, p, r, k)
2:     ▷ find the median of x[A[i]], and partition A using it
3:     $q \leftarrow$ Select$(A, p, r, \lfloor \frac{p+r}{2} \rfloor)$
4:     Swap$(A[q], A[r])$
5:     $q \leftarrow$ Partition(A, p, r)
6:     ▷ count the number of elements smaller than the median
7:     $sum \leftarrow 0$
8:     **for** $i \leftarrow p$ **to** $q-1$ **do**
9:         $sum \leftarrow sum + w[A[i]]$
10:     **if** $sum < k \leq sum + w[A[q]]$ **then**
11:         **return** $x[A[q]]$
12:     **else if** $k \leq sum$ **then**
13:         **return** SelectPairs$(A, p, q-1, k)$
14:     **else**
15:         **return** SelectPairs$(A, q+1, r, k - sum - w[A[q]])$

בשורות 3-9 מספר הפעולות שאנו מבצעים הינו לינארי ביחס ל-n. הקריאות הרקורסיביות בשורות 13 ו-15 רצות על חצי מגודל הקלט המקורי, סה"כ נקבל את נוסחת הנסיגה:

$$T(n) = T\left(\frac{n}{2}\right) + \Theta(n) = T(1) + \sum_{k=0}^{\lfloor \lg n \rfloor - 1} \Theta(\frac{n}{2^k})$$

$$= \Theta(1) + \Theta(n \underbrace{\sum_{k=0}^{\lfloor \lg n \rfloor - 1} \frac{1}{2^k}}_{\leq 2}) = \Theta(n)$$