1) א. נסמן את שם האלגוריתם ב- NMS. יקבל מערך A ואת אורך המערך n.

הקוד:

```
NMS(A, N)
    L = 1
    R = n
    while(L+1 < R):
        m = ⌊(L+R)/2⌋
        if A[m] > A[0]:
            R = m-1
        else:
            L = m
    if A[R] = A[0]:
        return R
    else:
        return L
```

זמן הריצה יהיה בסדר גודל של $\log n$, שכן בכל איטרציה של הלולאה, נחצה את המערך.

נכונות: האלגוריתם מחפש את הערך הכי ימני במערך של המספר המינימלי.

ברגע שנגיע למערך בגודל שקטן מ-2, נבדוק האם הערך הימני הוא המינימום.

אם כן, נחזיר את האינדקס שלו. אחרת, הערך המינימלי מופיע בתא השמאלי ונחזיר את

האינדקס שלו.

ב. זה קוד שמחזיר את n_{max}.

- כל עוד יש מופע אחד של המקסימום, לא נכנס ללולאה ה-1 ונחזיר 1. זה הגיוני כי זה אומר שהמקסימום מופיע רק במקום ה-n ולא משמאלו.

- אחרת, המשך הקוד מחפש את המופע השמאלי ביותר של המקסימום במערך, ומחזיר את אורך תת המערך מהמקום השמאלי בו מופיע המקסימום ועד n.

<u>זמן ריצה:</u>

$k < \log n$

הלולאה הראשונה: $K = 2^h < \log n$ $\Rightarrow$ בלולאה הראשונה $\log(\log n)$ פעולות

$h < \log(\log n)$

פעולות בודדות, $O(1)$.

הלולאה השנייה: $max \geq n - \frac{\log n}{2}$

$min \sim n - \log n + 1$

$\Rightarrow max - min \sim \frac{\log n}{2} - 1$

$\Leftarrow$ אלגוריתם של חיפוש בינארי על רשימה בסדר גודל של $\log n \Rightarrow \log(\log n)$.

לכן, זמן הריצה יהיה מסדר גודל $\underline{\log(\log n)}$.

2) א. i. not found.

ii. הנחנו בשלילה שקיים אלגוריתם שדורש לכל היותר n-1 צעדים. עבור מערך באורך n, קיים לפחות תא אחד (במיקום ה-i) אליו האלגוריתם לא יפנה.

ב. i. הרצת האלגוריתם תהיה זהה. באלגוריתם שרץ על A בדקנו בכל התאים מלבד בתא ה-i. בריצה על B אותו דבר, ללא חשיבות הערך בתא ה-i שהשתנה.

ii) not found - סתירה.

ג. שיטה זו לא תעבוד כי הנחנו בשלילה שקיים אלגוריתם לחיפוש במערך לא ממוין שדורש פחות מ- n-1 צעדים והגענו לסתירה, ועבור מערך ממוין, ידוע שקיים כזה אלגוריתם (חיפוש בינארי).

3) א.

A_1	4	3	2	1	1	1	1	1
A_2	4	4	3	3	2	2	1	1

$$T_{IS}(A_1) = \sum_{j=1}^{n} t_j(A) = \sum_{j=1}^{n^{\frac{2}{3}}} j + \sum_{j=n^{\frac{2}{3}}+1}^{n} n^{\frac{2}{3}}$$

$$\frac{(1+n^{\frac{2}{3}})(n^{\frac{2}{3}}-1)}{2} + n^{\frac{2}{3}}\cdot(n-n^{\frac{2}{3}}) =$$

$$= \frac{\cancel{n^{\frac{2}{3}}} - 1 + n^{\frac{4}{3}} - \cancel{n^{\frac{2}{3}}}}{2} + n^{\frac{5}{3}} - n^{\frac{4}{3}} = n^{\frac{5}{3}} - \frac{n^{\frac{4}{3}}}{2} - \frac{1}{2}$$

$$T_{IS}(A_2) = \sum_{j=1}^{n} \underbrace{t_j(A)}_{j-1} = \frac{(n-1)(n-1)}{2} = \frac{(n-1)^2}{2}$$

ב. 0

ג. $j-1$

ד. יש $L_A(j)$ איטרציות בתוך ה-while באיטרציה ה-j.

ה. $T_{IS}(A) = \sum_{j=1}^{n} C_1 + C_2 L_A(j)$

ביחס לסעיף ב - המערך ממוין ו- $L_A(j)=0$ לכל j לכן $T_{IS}(A_{\text{ממוין}}) = C_1 n$ כפי שראינו.

ביחס לסעיף ג - המערך ממוין הפוך ו- $L_A(j)=j-1$ לכן, $T_{IS}(A_{\text{ממוין הפוך}}) = \Theta(n^2)$ כפי שראינו.

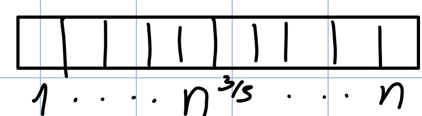
ו. (i) לא יתכן. הראינו בהרצאה שהחסם התחתון לזמן הריצה הוא $\Omega(n)$.

כאן החסם העליון קטן מהחסם התחתון שהראינו.

(ii) לא יתכן. הראינו בהרצאה שהחסם העליון לזמן הריצה הוא $O(n^2)$.

כאן החסם התחתון גדול מהחסם העליון שהראינו.

(iii) יתכן. ניקח $n^{3/5}$ מספר שלם כלשהו. עבור מערך A באורך n מהצורה



בו כל הערכים במקומות 1 עד $n^{3/5}$ ממוינים מהגדול לקטן, ועבור כל שאר המקומות, הערכים ממוינים מקטן לגדול ובנוסף מתקיים:

$A[j] > A[i]$, $1 \le i \le n^{3/5}$, $n^{3/5}+1 \le j \le n$.

זמן הריצה של תת המערך שממוין הפוך יהיה מסדר גודל של $\Theta(n^{6/5})$.

ושל תת המערך הממוין יהיה מסדר גודל של $\Theta(n)$. ובסה"כ, $C' n^{6/5} \le T_{IS}(A) \le C n^{6/5}$.

4) (1) נכון בהכרח. $\Omega(n^{\frac{1}{4}})$ קטן מכל החסמים התחתונים שנתונים לפונקציות השונות.
לכן, זמן הריצה יהיה גדול גם מהחסם הזה.

(2) לא בהכרח נכון. מוגדר $T_2(n) = O(n^2)$ אבל אם זה לא חסם הדוק, הסיבוכיות
יכולה להיות קטנה מ-n^2 ואז זה לא יתקיים.

(3) בהכרח לא נכון. עבור $T_1(n) = \Omega(n \log^3 n)$, מתקבל שזמן הריצה הכולל
גדול מהחסם $O(n)$.

5) א. $A = [2, 4, 7, 8, 1, 3, 5, 6]$

ב. החסם העליון הוא $m \leq \frac{n}{2^i} - 1$ מס' האינדקסים עבורם $A[x] > A[x+1]$

נסביר: בסיום האיטרציה ה-i, יהיו $\frac{n}{2^i}$ תתי מערך ממוינים באורך 2^i
כל אחד. לכן, יתכנו $\frac{n}{2^i} - 1$ זוגות תאים עוקבים רק ש-$A[x] > A[x+1]$.

ג. תשובה (2).

נסביר: Alg מפרק את כל המערך ליחידות, ממזג לזוגות ואז לרביעיות
וכן הלאה. אלג' MS תחילה ממיין את תת המערך השמאלי ורק אז את
הימני.

$$T_{1st\,loop} = \sum_{j=0}^{n/k-1} t_j = \sum_{j=0}^{n/k-1} T_{IS(k)} \;\begin{array}{l} \xrightarrow{\text{best case}} cn \\ \xrightarrow[\text{worst case}]{} c'nk \end{array}$$ .3

$$\sum_{j=0}^{\frac{n}{2^i}-1} T_{merge(j)} = \sum_{j=0}^{\frac{n}{2^i}-1} 2^i = 2^i\left(\frac{n}{2^i}-1+1\right) = n$$

$$T_{1st\,for} = n\cdot\left(\log(n)-\log(k)-1+1\right) = n\cdot\log\left(n/k\right)$$

ה. עבור $k=1$, זמן הריצה הוא $n\log(n)$

עבור $k=n$, זמן הריצה הוא של IS (לא נכנס ל- Merge).

לכן, התשובה היא כן. עבור מערך A ממוין מהקטן לגדול, עבור $k=1$ זמן הריצה $n\log n$.

עבור $k=n$, זמן הריצה הוא של IS על מערך ממוין, ולכן זמן ריצה של n, ו- $n < n\log n$.