

מה"ט Java – סיבוכיות זמן ריצה

מהי סיבוכיות זמן ריצה?

(n) סיבוכיות זמן מתארת את מספר הפעולות שהקוד מבצע ביחס לגודל הקלט.

אלא אם נאמר אחרת, (Worst Case), מדובר במקרה הגרוע ביותר.

Big-O מסמנים בעזרת (כמו $O(n)$, $O(n^2)$, $O(\log n)$)

דוגמאות נפוצות

for (int i = 0; i < n; i++) → $O(n)$ – לולאה אחת

for (int i = 0; i < n; i++) for (int j = 0; j < n; j++) → $O(n^2)$ – לולאות מקוננות

while (n > 0) { n /= 10; } → $O(\log n)$ – n מספר הספרות של

if (...) return ... → $O(1)$ – פעולה אחת קבועה

דוגמה משאלה אמיתית

```
for (int i = 0; i < arr.length; i++)
```

```
    for (int j = i + 1; j < arr.length; j++)
```

```
        if (arr[i] == arr[j]) return false;
```

```
return true;
```

עבור כל איבר אנו עוברים על שאר האיברים – $O(n^2)$ הסיבוכיות היא →

שאלות מה"ט טיפוסיות

$O(n)$ שאלות עם לולאות בודדות → לרוב

$O(n^2)$ שאלות עם בדיקות כפולות על מערכים → לרוב

$O(\log n)$ בדיקות ספרות במספר → לרוב

$O(1)$ בדיקות תנאי פשוט →

סיכום טבלה

$O(n)$ → לולאה בודדת על מערך

$O(n^2)$ → שתי לולאות מקוננות

$O(\log n) \rightarrow$ לולאה עם חלוקה (ב-2 או ב-10)

$O(1) \rightarrow$ return / תנאי פשוט