

מעריך שיעור – יעילות

דפנה מינסטר

מטרת השיעור:

הטמעת נושא היעילות אצל התלמידים.

מטרות משנה:

1. התלמיד/ה יפגינו הבנה של תלות סדר הגודל של פסאודו-קוד (אלגוריתם) נתון כתלות של אורך הקלט.
2. בהינתן פסאודו-קוד (אלגוריתם) כלשהו וקלטים שונים, התלמיד/ה יזהו את המקרה הגרוע ביותר במונחי סדר-גודל.
3. התלמיד/ה יסבירו את המושגים הבאים: "אורך קלט", "צעד בסיסי", "פעולות תלויות באורך הקלט", "פעולות שאינן תלויות באורך הקלט".

זמן השיעור: שיעור (45 דקות).

ידע קודם:

- ❖ יסודות $1 + 2$
- ❖ רקורסיה.
- ❖ הפעולות: מיון לסוגיו, חיפוש בינארי ומיזוג.

מהלך השיעור:

פתיחה: (5 דקות) חלוקת דף העבודה, העבודה היא אישית.

גוף השיעור:

1. (18-20 דקות) פתרון דף העבודה (בהמשך) ע"י כל אחד מתלמידי הכיתה.
2. (15 דקות) פתרון דף העבודה (בהמשך) במליאת הכיתה בהנחיית המורה.

סיכום השיעור: (5-7 דקות)

סיכום הפתרונות שהועלו בכיתה ונושא היעילות ע"י המורה, תוך כדי שאילת שאלות מנחות בכיתה, לדוגמה: הסבר המושגים "אורך קלט", "צעד בסיסי", "פעולות תלויות באורך הקלט", "פעולות שאינן תלויות באורך הקלט".

בהצלחה!

יעילות - בלולאות

מס'	אלגוריתם	סיבוכיות זמן הריצה (סדר גודל), ונימוק!
1.	<pre>int n = read(); int sum; if (n % 2 == 0) sum = n; else { sum = 0; for (int i=1; i ≤ n; i++) sum = sum + i; }</pre>	<hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>
2.	<pre>int n = read(); int sum=0; for (int i=1; i ≤ n; i++) for (int j=i; j ≤ n; j++) sum = sum + i + j;</pre>	<hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>
3.	<pre>for (int i=1; i ≤ 10; i++) { for (char tav = 'א'; tav ≤ 'ד'; tav++) System.out.print (tav + " התו: " + i + " השורה"); System.out.print (); }</pre>	<hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>
4.	<pre>int n = read(); for (int i=1; i ≤ n; i++) { for (char tav = 'א'; tav ≤ 'מ'; tav++) System.out.print (tav + " התו: " + i + " השורה"); System.out.print (); }</pre>	<hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>
5.	<pre>int max=0; for (int i=1; i ≤ n; i++) for (int j=i; j ≤ n; j++) { num = read (); if (max < num) max = num ; }</pre>	<hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>
6.	<pre>int n = read(), m = read(); int sum=0; for (int i=1; i ≤ n; i++) { for (int j = i ; j ≤ m; j++) sum = sum + i + j; }</pre>	<hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>
7	int i=1:	



	<pre>int count=0; while (i ≤ 100) { for (int j=i; j ≤ n2 ; j++) count++; i = i * 2; }</pre>	
8.	<pre>int x = 10; for (int i=1 ; i ≤ x ; i++) { for (char tav = 'א'; tav ≤ 'מ'; tav++) System.out.print (tav + " :התו" + i + " :השורה"); System.out.print () ; }</pre>	
9.	<pre>int n = 2; int i=1; int sum=0; while (i ≤ n) { sum += i * 2 ; n = n * 2 ; i++; }</pre>	
10.	<pre>int n = read() ; int i=1; int count=0; while (i ≤ n) { for (int j=i; j ≤ n² ; j++) count++; i = i * 2; }</pre>	
11.	<pre>func1 (n,m) { int sum=1; for (int i=1; i ≤ n; i++) { for (int j=1; j ≤ m ; j++) sum = sum + func2 (i + j); } } func2 (x) { int sum=0; for (int i=1 ; i ≤ x² ; i++) sum += i; }</pre>	



יעילות - בלולאות

מס'	אלגוריתם	סיבוכיות זמן הריצה (סדר גודל), ונימוק!
1.	<pre>int n = read(); int sum; if (n % 2 == 0) sum = n; else { sum = 0; for (int i=1; i ≤ n; i++) sum = sum + i; }</pre>	<p>הסדר גודל $O(n)$ - n מציין את מספר המספרים (1-n).</p> <p>במקרה הגרוע, אם המספר הוא אי-זוגי, ייסכמו המספרים מ-1 ועד n.</p>
2.	<pre>int n = read(); int sum=0; for (int i=1; i ≤ n; i++) for (int j=i; j ≤ n; j++) sum = sum + i + j;</pre>	<p>הסדר גודל $O(n^2)$ - n מציין את מספר המספרים (1-n).</p> <p>עבור כל מספר, האלגוריתם יעבור שוב על כל המספרים מ-i ועד n.</p> <p>$n^2 \Leftrightarrow \frac{n^2}{2}$</p>
3.	<pre>for (int i=1; i ≤ 10; i++) { for (char tav = 'א'; tav ≤ 'ד'; tav++) System.out.print (tav + "התו: " + i + "השורה: "); System.out.print (); }</pre>	<p>הסדר גודל $O(1)$</p> <p>כל אחת מהלולאות המקוננות פועלת מספר קבוע של פעמים (מספר הפעולות הוא 130).</p>
4.	<pre>int n = read(); for (int i=1; i ≤ n; i++) { for (char tav = 'א'; tav ≤ 'מ'; tav++) System.out.print (tav + "התו: " + i + "השורה: "); System.out.print (); }</pre>	<p>הסדר גודל $O(n)$</p> <p>n - מציין את מספר המספרים (1-n).</p> <p>עבור כל מספר מ-1 ועד n (לולאה חיצונית), תופעל הלולאה הפנימית מספר קבוע של פעמים (13 פעמים).</p>
5.	<pre>int n = read(); int max=0; for (int i=1; i ≤ n; i++) for (int j=i; j ≤ n; j++) { num = read(); if (max < num) max = num; }</pre>	<p>הסדר גודל $O(n^2)$ - n מציין את מספר המספרים (1-n).</p> <p>עבור כל מספר, האלגוריתם יעבור שוב על כל המספרים מ-1 ועד n לצורך חישוב הערך המקסימלי.</p>
6.	<pre>int n = read(), m = read(); int sum=0;</pre>	<p>הסדר גודל $O(n*m)$ - ממשפחת $O(n^2)$</p> <p>n - מציין את מספר המספרים (1-n).</p>



	<pre>for (int i=1 ; i ≤ n ; i++) { for (int j = i ; j ≤ m ; j++) sum = sum + i + j; }</pre>	<p>m – מציין את מספר המספרים (1-m).</p> <p>עבור כל מספר מ-n, האלגוריתם יעבור שוב על כל המספרים מ-1 ועד m. לצורך חישוב הסכום.</p>
7.	<pre>int n = read(); int i=1; int count=0; while (i ≤ 100) { for (int j=i; j ≤ n²; j++) count++; i = i * 2; }</pre>	<p>הסדר גודל $O(n^2)$ – מציין את מספר המספרים (1-n).</p> <p>יש 3 לולאות מקוננות, הלולאה החיצונית (while) פועלת מספר קבוע של פעמים. בשתי הלולאות המקוננות הפנימיות האחרות, עבור כל מספר, האלגוריתם יעבור שוב על כל המספרים מ-1 ועד n.</p>
8.	<pre>int x = 10; for (int i=1 ; i ≤ x ; i++) { for (char tav = 'א' ; tav ≤ 'מ' ; tav++) System.out.print (tav + "התו: " + i + "השורה: "); System.out.print (); }</pre>	<p>הסדר גודל $O(1)$</p> <p>כל אחת מהלולאות המקוננות פועלת מספר קבוע של פעמים (מספר הפעולות הוא 130). המשתנה x הוא קבוע על 10.</p>
9.	<pre>int n = 2; int i=1; int sum=0; while (i ≤ n) { sum += i * 2; n = n * 2; i++; }</pre>	<p>הסדר גודל - לא פתיר</p> <p>הלולאה תתבצע אינסוף פעמים – לא תעצור.</p>
10.	<pre>int n = read(); int i=1; int count=0; while (i ≤ n) { for (int j=i; j ≤ n²; j++) count++; i = i * 2; }</pre>	<p>הסדר גודל $O(n^2 \cdot \log(n))$ – מציין את מספר המספרים (1-n).</p> <p>יש 3 לולאות מקוננות, הלולאה החיצונית (while) קטנה כל פעם פי 2 – $\log(n)$. בשתי הלולאות המקוננות הפנימיות האחרות, עבור כל מספר, האלגוריתם יעבור שוב על כל המספרים מ-1 ועד n.</p>
11.	<pre>func1 (n,m) { int sum=1;</pre>	<p>הסדר גודל $O(n \cdot m(n+m)^2)$ – ממשפחת $O(n^4)$</p> <p>m – מציין את מספר המספרים (1-n).</p>



<pre> for (int i=1; i ≤ n; i++) { for (int j=1; j ≤ m ; j++) sum = sum + func2 (i + j); } </pre>	<p>m – מציין את מספר המספרים (1-m).</p> <p>עבור כל מספר מ- n, האלגוריתם יעבור שוב על כל המספרים מ- 1 ועד m, ויזמן את הפעולה func2 שסדר הגודל שלה $O(x^2)$.</p>
<pre> func2 (x) { int sum=0; for (int i=1 ; i ≤ x² ; i++) sum += i ; } </pre>	<p>func2 (x)</p> <p>הסדר גודל $O(x^2)$ – מציין את מספר המספרים (1-x²).</p> <p>עבור כל מספר, האלגוריתם יעבור שוב על כל המספרים מ- 1 ועד x².</p>

