

## Power

הבעיה: מציאת אלגוריתם דחוש, חזקה בסיבוכיות של  $O(\log_2 n)$  ?

$x^n$

צריך דעצות מספר ממשי  $x$  בחזקת  $n$ . נעביר את מספר  $n$  להצגה בינארית:

$$m_i = \{0, 1\}, i = 0, \dots, k \text{ כאשר } n = m_k 2^k + m_{k-1} 2^{k-1} + m_{k-2} 2^{k-2} + \dots + m_1 2 + m_0$$

$$x^m = x^{m_0} \cdot (x^2)^{m_1} \cdot (x^{2^2})^{m_2} \cdot (x^{2^3})^{m_3} \cdot \dots \cdot (x^{2^k})^{m_k}$$

מכאן נקבל כי:

מימוש האלגוריתם באמצעות: אנו צריכים בכל שלב, דגמור את חזקה של מספר  $x$ ,

אך דבצע כפל רק כאשר בהצגה בינארית של המספר  $m_i \neq 0$ .

דוגמה:

$$\text{א) } n=7, \text{ ההצגה בינארית: } n = 111, \text{ דכן } 7 = 2^0 + 2^1 + 2^2 \rightarrow x^7 = x \cdot x^2 \cdot x^4$$

$$\text{ב) } n=6, \text{ ההצגה בינארית: } n = 110, \text{ דכן } 6 = 2^1 + 2^2 \rightarrow x^6 = x^2 \cdot x^4$$

$$\text{ג) } n=10, \text{ ההצגה בינארית: } n = 1010, \text{ דכן } 10 = 2^1 + 2^3 \rightarrow x^{10} = x^2 \cdot x^8$$

כס"א

קובץ:

```
public class Power {
```

```
    static double powerLoop(double number , int power ){
        double result = 1;
        while( power > 0 ){
            if ( power%2 == 1 ) // אם החזקה אי-זוגית
                result = result * number;
            number = number * number;
            power = power / 2;
        }
        return result;
    }
```

עבד את הפתרון עם הוויזואל  
מאשר את הפתרון הרקורסיבי.

```
    static double powerRecursion(double number, int power){
        if ( power == 0 )
            return 1;
        else
            if ( power%2 == 0 )
                return powerRecursion( number*number , power/2 );
            else
                return number * powerRecursion( number*number , power/2 );
    }
```

```
public static void main(String[] args) {
```

```
    for( int i =1; i<= 10 ; i++ ){
        System.out.print( "2^"+i+" = "+(int)powerLoop(2, i) );
        System.out.println( " " + (int)powerRecursion(2, i) );
    }
```

```
}
```

סיבוכיות:  $O(\log_2 n)$