**שיעורי בית יסודות מיון בחירה ומיון בועות – אופיר הופמן י3**

**מכתיבת הקוד מצאתי ש:**

**במיון בחירה:**

1. **כאשר הגדלתי את המערך פי 5, זמן המיון עלה פי 10.**
2. **מיון מערך ממויין מראש לקח פי 10 מהר יותר מאשר מערך רגיל (כמו סעיף 1)**

**במיון בועות:**

1. **כאשר הגדלתי את המערך פי 5, זמן המיון עלה פי 25.**
2. **מיון מערך ממויין מראש כמעט ולא לקח שום זמן, מלבד ריצה אחת על המערך.**

**\*במיון בועות ללא תנאי עצירה – הוא רץ באופן רגיל, ולכן לוקח יותר זמן ופחות יעיל.**

* **מסקנה נוספת: ככל שעולה גודל המערך, כך ההפרש בין זמן מיון בשיטת בועות לבין זמן המיון בשיטת בחירה עולה – תלוי באופן שבו ממויין המערך! (אם המערך ממויין אז לא)**
* **במיון בועות ללא תנאי עצירה: ההפרש גדל אפילו יותר.**

public static void PrintArr(int[] arr)

{

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

{

Console.Write(arr[i] + "|");

}

Console.WriteLine();

}

public static void SelectionSort(int[] arr)

{

int temp, pmin;

for(int i = 0; i < arr.Length - 1; i++)

{

pmin = i;

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

{

if (arr[j] < arr[pmin])

pmin = j;

}

if (pmin != i)

{

temp = arr[i];

arr[i] = arr[pmin];

arr[pmin] = temp;

}

}

}

public static void BubbleSort(int[] a)

{

int temp;

bool sorted = false;

for (int i = 0; i < a.Length - 1 && !sorted; i++)

{

sorted = true;

for (int j = 0; j < a.Length - i - 1; j++)

{

if (a[j] > a[j + 1])

{

temp = a[j];

a[j] = a[j + 1];

a[j + 1] = temp;

sorted = false;

}

}

}

}

static void Main(string[] args)

{

// simple array selection sort

int[] selectionArr = { 4, 2, 1, 5, 3 };

PrintArr(selectionArr);

SelectionSort(selectionArr);

PrintArr(selectionArr);

// simple array bubble sort

int[] BubbleArr = { 4, 2, 1, 5, 3 };

PrintArr(BubbleArr);

BubbleSort(BubbleArr);

PrintArr(BubbleArr);

int[] bigSelArr = new int[10000];

Random rnd = new Random();

for (int i = 0; i < bigSelArr.Length; i++)

{

bigSelArr[i] = rnd.Next(1, 1001);

}

int[] bigBubArr = new int[10000];

for (int i = 0; i < bigBubArr.Length; i++)

{

bigBubArr[i] = bigSelArr[i];

}

Console.ReadKey();

//Check run time Big arrays

//selection

DateTime selectionBefore = DateTime.Now;

SelectionSort(bigSelArr);

DateTime selectionAfter = DateTime.Now;

Console.WriteLine("Selection:");

Console.Write(selectionAfter-selectionBefore);

Console.WriteLine();

//Bubble

DateTime bubbleBefore = DateTime.Now;

BubbleSort(bigBubArr);

DateTime bubbleAfter = DateTime.Now;

Console.WriteLine("Bubble:");

Console.WriteLine(bubbleAfter-bubbleBefore);

//five times bigger arrays

int[] biggestSelArr = new int[bigSelArr.Length\*5];

for (int i = 0; i < biggestSelArr.Length; i++)

{

biggestSelArr[i] = rnd.Next(1, 1001);

}

int[] biggestBubArr = new int[bigBubArr.Length\*5];

for (int i = 0; i < biggestBubArr.Length; i++)

{

biggestBubArr[i] = biggestSelArr[i];

}

Console.ReadKey();

// Check run time of a 5 times bigger arrays

// selection

DateTime selectionBefore2 = DateTime.Now;

SelectionSort(biggestSelArr);

DateTime selectionAfter2 = DateTime.Now;

Console.WriteLine("Selection:");

Console.Write(selectionAfter2 - selectionBefore2);

Console.WriteLine();

// bubble

DateTime bubbleBefore2 = DateTime.Now;

BubbleSort(biggestBubArr);

DateTime bubbleAfter2 = DateTime.Now;

Console.WriteLine("Bubble:");

Console.Write(bubbleAfter2 - bubbleBefore2);

Console.WriteLine();

// sorted arrays

int[] sortedSelArr = new int[10000];

for (int i = 0; i < sortedSelArr.Length; i++)

{

sortedSelArr[i] = i;

}

int[] sortedBubArr = new int[10000];

for (int i = 0; i < sortedBubArr.Length; i++)

{

sortedBubArr[i] = i;

}

Console.ReadKey();

// Check run time of sorted arrays

// selection

DateTime selectionBefore3 = DateTime.Now;

SelectionSort(sortedSelArr);

DateTime selectionAfter3 = DateTime.Now;

Console.WriteLine("Selection: ");

Console.Write(selectionAfter3 - selectionBefore3);

Console.WriteLine();

//bubble

DateTime bubbleBefore3 = DateTime.Now;

BubbleSort(sortedBubArr);

DateTime bubbleAfter3 = DateTime.Now;

Console.WriteLine("Bubble: ");

Console.Write(bubbleBefore3);

Console.WriteLine(bubbleAfter3 – bubbleBefore3);

Console.ReadKey();

}