**ממן 14 – מבני נתונים ומבוא לאלגוריתמים – ברנדס איתי ת.ז.**

**סעיף א':**

נמצא בקובץ המצורף – mmn14.py. נא להריץ בעזרת פייטון 2.7.

**סעיף ב':**

בעת הרצת התוכנית, קיבלתי כפלט:

Creating a 50 elements random array as A.

Creating a 100 elements random array as B.

Creating a 200 elements random array as C.

Running heapsort(A, 2); Total Comparisons: 420 Total Copies: 729

Running heapsort(B, 2); Total Comparisons: 1022 Total Copies: 1764

Running heapsort(C, 2); Total Comparisons: 2445 Total Copies: 4089

Running heapsort(A, 3); Total Comparisons: 412 Total Copies: 537

Running heapsort(B, 3); Total Comparisons: 1011 Total Copies: 1251

Running heapsort(C, 3); Total Comparisons: 2391 Total Copies: 2895

Running heapsort(A, 4); Total Comparisons: 440 Total Copies: 474

Running heapsort(B, 4); Total Comparisons: 1064 Total Copies: 1068

Running heapsort(C, 4); Total Comparisons: 2595 Total Copies: 2496

Running heapsort(A, 5); Total Comparisons: 487 Total Copies: 426

Running heapsort(B, 5); Total Comparisons: 1185 Total Copies: 978

Running heapsort(C, 5); Total Comparisons: 2783 Total Copies: 2196

ביצועי האלגוריתם heapsort שקיבלתי בעת הרצת סעיף א' הינם:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | מספר השוואות | | | | מספר העתקות | | | |
| **d=2** | **d=3** | **d=4** | **d=5** | **d=2** | **d=3** | **d=4** | **d=5** |
| A (מערך בגודל 50) | 420 | 412 | 440 | 487 | 729 | 537 | 474 | 426 |
| B (מערך בגודל 100) | 1022 | 1011 | 1064 | 1185 | 1764 | 1251 | 1068 | 978 |
| C (מערך בגודל 200) | 2445 | 2391 | 2595 | 2783 | 4089 | 2895 | 2469 | 2196 |

לפי הנתונים לעיל ניתן להסיק כי:

1. מבחינת מספר ההשוואות, האלגוריתם יעיל ביותר עבור d=3.
2. מבחינת מספר ההעתקות, האלגוריתם יעיל ביותר עבור d=5.

ז"א, כאשר זמן הריצה של השוואה קצר מזמן הריצה של העתקה, נעדיף להשתמש בערמה טנארית (d=3). כאשר זמן הריצה של השוואה ארוך מזמן הריצה של העתקה, נעדיף להשתמש בערמה d-ית כאשר d=5.

המצב השני עלול להתרחש כאשר ה-Cache במעבד מלא, ושאף הRAM מלא, ואז ע"מ לבצע השוואות באופן יעיל יש לטעון את המערך לזיכרון, אך מפאת חוסר מקום הוא יטען לכונן הקשיח, שזמן הקריאה אליו הוא ארוך משמעותית. נראה שבמצב זה, כאשר המערכת עמוסה / הערמה גדולה מדי ע"מ להיטען לזיכרון הCache כולו, נקבל ביצועי השוואות נחותים, ואז נעדיף להשתמש בערמה d-ית בעל ערך d גדול.