

דוגמה לפלט התוכנית:

```
$ for i in {1..3}; do python sort.py; echo; done
bubblesort did 19900 comparsions, 29676 copies on average
bidibubblesort did 15487 comparsions, 29676 copies on average
shakesort did 20000 comparsions, 600 copies on average

bubblesort did 19900 comparsions, 29498 copies on average
bidibubblesort did 15406 comparsions, 29498 copies on average
shakesort did 20000 comparsions, 600 copies on average

bubblesort did 19900 comparsions, 29593 copies on average
bidibubblesort did 15393 comparsions, 29593 copies on average
shakesort did 20000 comparsions, 600 copies on average
```

**BubbleSort** זמן הריצה במקרה הממוצע (והטוב) של האלגוריתם הוא  $\Theta(n^2)$ , דבר שמתבטא במספר ההשוואות שמבצע האלגוריתם. מספר ההעתיקות גם כן בסדר גודל של  $\Theta(n^2)$  שכן המעבר ה- $i$  של הלולאה החיצונית ממקם את האיבר ה- $i$  בגודלו במקום שלו במערך הממויין ע"י החלפתו עם האיבר מימינו.

**BidiBubble** התוצאות באלגוריתם זה דומות לתוצאות שראינו במיון בועות, וגם כאן סיבוכיות האלגוריתם במקרה הממוצע היא  $\Theta(n^2)$ . מספר ההשוואות הנמוך יחסית נובע מכך שהאלגוריתם עוצר אם היה מעבר על המערך שלא ביצע שום החלפה. בנוסף, האלגוריתם מטפל גם באיבר המינימלי בכל איטרציה של הלולאה הראשית. ביצועיו טובים יותר על מערכים "כמעט" ממויינים. לדוגמה המערך  $[2, 3, 4, 1]$  ימויין לאחר מעבר אחד, לעומת שלושה מעברים במיון בועות רגיל.

אלגוריתם זה הוא היחיד מבין השלושה שזמן הריצה שלו במקרה הטוב הוא לינארי.

**ShakeSort** בכל שלב האלגוריתם מוצא את המקסימום והמינימום בתת מערך של המערך המקורי ולכן גם במקרה הטוב זמן הריצה של האלגוריתם הינו  $\Theta(n^2)$ . זה עומד בקנה אחד עם מספר ההשוואות שהאלגוריתם מבצע שכן בפלט התוכנית אנחנו רואים שהוא אינו תלוי בסדר האיברים במערך ולכן אינו משתנה בין הרצות. מצד שני מבחינת העתקות האלגוריתם אופטימלי ביחס לקודמיו שכן כל איבר מועתק פעם אחת בלבד למיקום הסופי שלו במערך הממויין.