

ממ"ן 13 - מבני נתונים ומבוא לאלגוריתמים - עומר קומיסרצ'יק - 215314725

n	4	5	6	7	7	8	8	9	10	11
a	0.33	0.20	0.20	0.33	0.01	0.50	0.01	0.01	0.001	0.0001
Quick sort - best average	4.00	6.00	8.44	11.28	10.00	14.73	13.00	16.00	19.00	22.00
Quick sort – average	4.83	7.40	10.30	13.48	13.48	16.92	16.92	20.58	24.44	28.48
Quick sort - worst average	6.00	9.67	13.22	16.28	21.00	19.11	26.75	32.45	42.48	53.41
Best pivot quick sort - best average	16.00	24.00	32.67	56.68	36.00	66.37	47.00	58.00	69.00	80.00
Best pivot quick sort – average	21.33	44.00	51.62	90.00	90.00	89.61	89.61	140.67	146.83	150.31
Best pivot quick sort – worst average	28.57	64.00	75.25	123.32	144.00	112.85	158.97	228.47	280.70	339.58

הקוד שלי נכתב בשפת C, השתמשתי לצורך הכתיבה ב ubuntu כפי שעושים בקורס מעבדה בתכנות מערכות, צירפתי קובץ ריצה וגם makefile.

ננתח סיבוכיות לכל אלגוריתם מיון:

A הוא מערך מספרים שלמים בגודל n.

Quick sort: (לפי הספר)

- סיבוכיות זמן ריצה במקרה הטוב $O(n \lg n)$

- סיבוכיות זמן ריצה במקרה הממוצע $O(n \lg n)$

- סיבוכיות זמן ריצה במקרה הגרוע $O(n^2)$

Best pivot quick sort: (כפי שניסחנו)

בשונה מ quick sort הרגיל, האלגוריתם רוצה לבחור את ה pivot באופן כזה שהוא יחלק את מערך הפרמוטציה לשני מערכים שווים גודל (בשאיפה ל $n/2$) ולכן ה pivot הטוב ביותר הוא אמצע הפרמוטציה, ולכן אנחנו משתמשים בשיטה שבודקת לכל איברי המערך, עבור איזה איבר הפרש מספר האיברים שגדולים ממנו מהקטנים ממנו הוא לכל היותר 1 (וזהו האמצע למעשה) ואת ה pivot שמצאנו מעבירים ל partition ולאחר החלוקה מבצעים קריאת רקורסיה פעמיים על חלקי המערך בעלי הגודל הזהה ששואף ל $n/2$. יעילות השיטה למציאת האמצע היא במקרה הטוב $O(1)$, במקרה הגרוע $O(n^2)$ (אם הלולאות המקוננות ביעילות $O(n)$ כל אחת) ובמקרה הממוצע ביעילות גם כן של $O(n^2)$ כי במקרה הממוצע כל לולאה תגיע בערך לאמצע המערך ויעילות של כל לולאה תהיה לינארית. היעילות של partition היא $O(n)$ לכל מקרה, ולכן נסיק שבמקרה הטוב יעילות ה

Best pivot quick sort היא $O(n \lg n)$ כי למעשה המצב הטוב ביותר הוא כמו ב-quick sort רגיל שקיבלנו חלוקה מיטבית (שני תתי המערך בגודל $n/2$) והיעילות $O(n \lg n)$. במקרה הממוצע והגרוע מאחר ש partition והשיטה למציאת האינדקס לא משתנות ביעילות נקבל את נוסחת הנסיגה: $T_n = 2 * T_{n/2} + O(n) + O(n^2)$, פיתוח הנוסחה נותן לנו ש $T_n = O(2n^2)$. היעילויות של אלגוריתם quick sort מתיישבת בצורה קצת רעה עם הפרמוטציות, ייתכן שזה בגלל הזנחה של קבועים בפונקציית היעילות וגם הזנחה של ספירת פעולות אלמנטריות מסוימות לדוגמא, עבור $n = 10$ ביחד עם $a = 0.001$ קיבלנו ב quick sort ש:

במקרה הטוב יש 19 השוואות, לעומת $n \lg n$ שנותן 33. במקרה הממוצע יש דיוק טוב יותר 24 השוואות לעומת $n \lg n$ ששוב נותן 33 השוואות, ובמקרה הגרוע מקבלים 42 לעומת n^2 שנותן 100.

היעילויות של פונקציית best pivot quick sort מתיישבות עם הממצאים גם כן בצורה קצת רעה ושוב ככל הנראה מכיוון שיש הזנחה פעולות מסוימות/קבועים. ניתן גם לחשוב שבשתי הפונקציות כאשר a מאוד קטן (לדוגמא 0.001 שבחרנו) מכיל ספקטרום צר של מקרים מאוד מאוד קיצוניים של מספר רב או מועט של השוואות שלא מעיד על היעילות עבור שאר הפרמוטציות בצורה טובה. עבור $n = 10$ ביחד עם $a = 0.001$ נקבל:

במקרה הטוב יש 69 השוואות לעומת $n \lg n$ שנותן 33. במקרה הממוצע יש 146 לעומת $2n^2$ שנותן 200, ובמקרה הגרוע שהוא גם כן $2n^2$ אנו מקבלים 280 השוואות.

ללא ספק ניתן לראות ששיטת quick-sort הרגילה עדיפה על שיטת ה best pivot quick sort ומכאן ניתן להסיק שעבור מציאת ה pivot הטוב ביותר שיחלק את המערך לתתי מערך בגודל מאוזן של $n/2$ יש מחיר גבוה מאוד ביחס לשיטה הרגילה, והוא מחיר המציאה של אותו איבר במערך שיהווה pivot לחלוקה המאוזנת.