מטלת מנחה 14

אל המנחה: הילה כליף

מגישים: חן גולובנקו 324825405

גלעד סמואל 206876856

***הסבר כללי על מטרת המחקר:***

מטרת המחקר היא להשוות יעילות שתי שגרות נתונות למציאה ומיון של k האברים הקטנים ביותר של מערך כלשהו.

הפתרון הראשון מתבצע בעזרת בניית ערמת מינימום מהמערך הנתון ואחר כך חילוץ האיבר המינימלי מהערמה k פעמים (כאשר בכל חילוץ של האיבר המינימלי מחזירים את הערמה להיות ערמת מינימום).

הפתרון השני מתבצע בעזרת מציאת האיבר ה- k הקטן ביותר עם Randomized-Select וחלוקת המערך סביבו (איברים שגדולים ממנו מימינו ואיברים שקטנים או שווים לו משמאלו) ולאחר מכן ביצוע QuickSort על k האיברים השמאליים על מנת לקבל אותם באופן ממוין.

לצורך מימוש הפרויקט כתבנו תוכנית בשפת ג'אווה שסופרת את ההשוואות שמבצע כל אלגוריתם עבור K ו-N נתונים.

כתבנו את התוכנית בסביבת העבודה Visual Studio Code גרסה 1.67

התוכנית בנויה כך שבראשה אנו מגדירים משתנים גלובלים, ולאחר מכאן מופיע ה- main שמריץ את התוכנית:

- מקבלת את המשתמש ומבקשת ממנו להקליד ערך ל- n ו- k   
 (לאחר ההזנה של הערך הרצוי יש ללחוץ Enter)

* לאחר מכן התוכנית שואלת את המשתמש אם הוא רוצה לבחור בעצמו את אברי המערך. במידה וכן עליו להקליד me, במידה ולא, עליו להקליד כל דבר אחר.
* לאחר מכן התוכנית מתחילה בשני שלבי הניסוי.  
  בשלב הראשון היא מפעילה את שני שגרות הניסוי של שני מערכים זהים בהתאם לערכי הn ו-k שהמשתמש הזין   
  בשלב השני היא מריצה אותן על 4 סדרות אקראיות. ומחזירה בהתאם לכמות ההשוואות שבוצעה איזו שגרה הייתה יעילה יותר.

***תוצאות המיני-מחקר:***

כזכור, מטרת הפרויקט הייתה לנתח איזה פתרון עדיף לבעיית מציאת k האיברים הקטנים במערך והחזרתם באופן ממוין.

***תוצאות:***

התוצאות שהתקבלו מההרצות בסעיף ב' אינן חד משמעיות. עבור k גדול עד כדי שווה ל n קיבלנו שאכן אלגוריתם הערמה משתמש בפחות השוואות מאשר סלקט עם מיון מהיר,

ואילו עבור k קטנים קיבלנו שסלקט ואז מיון משתמש בפחות השוואות כמעט ללא תלות בn , אך תוצאה זאת משתנה לעיתים בהתאם לערכים האקראיים אותם מקבל המערך.

בנוסף להרצות החובהת הרצנו גם עבור גודל סדרה 100,000 על מנת לבחון מה קורה בקלטים גדולים במיוחד ביחס לK וגם הוספנו הרצות עבור k=n לכל גדלי המערכים.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | n = 100 | n = 200 | n = 500 | n = 1000 | n = 100000 |
| k = 8 | Heap – 284  Select - 145 | Heap – 478  Select - 414 | Heap – 1035  Select - 1006 | Heap – 2014  Select - 2136 | Heap – 187,973  Select – 172,147 |
| k = 50 | Heap – 694  Select - 572 | Heap – 1011  Select - 1013 | Heap – 1697  Select - 1175 | Heap – 2764  Select - 2993 | Heap – 189421  Select - 149721 |
| k = 100 | Heap – 1026  Select - 1166 | Heap – 1608  Select - 1643 | Heap – 2463  Select - 3541 | Heap – 3640  Select - 5109 | Heap – 199148  Select - 186934 |
| k = n |  | Heap – 2471  Select - 4184 | Heap – 7443  Select - 14148 | Heap – 7447  Select - 12154 | Heap – 40,329,870  Select – 86,652,914 |

***ניתוח זמן ריצה תיאורטי של האלגוריתמים במקרה הגרוע הטוב והממוצע***

אלגוריתם ראשון – ערמה

לבניית ערמה יש חסם עליון של O(n) (עמוד 112).

לאחר מכן מבצעים heap-extract-min k פעמים לקבלת k האיברים הקטנים ביותר, כל ריצה של heap-extract לוקחת O(lgn) (עמוד 116) ולכן זמן הריצה הכולל של האלגוריתם במקרה הטוב ובמקרה הרע יהיה O(n+klgn) .

אלגוריתם שני – select ואז מיון

זמן ריצה במקרה הממוצע – לפי עמוד 157 זמן הריצה במקרה הממוצע של select עבור מערך עם איברים שונים הוא O(n) . נניח שהקלט מתפלג כמו שמופיע בעמוד זה בספר ולכן זמן הריצה יהיה זהה גם כאשר קיימים איברים זהים.

מיון מהיר במקרה הממוצע לוקח O(nlgn) (עמוד 126) ואנחנו מבצעים אותו על k איברים ולכן זמן הריצה של האלגוריתם במקרה הממוצע יהיה O(n+klgk) .

זמן ריצה במקרה הטוב – select בכל מקרה מבצעת מעבר ליניארי אחד לפחות ולכן זמן הריצה שלה גם במקרה הטוב יהיה O(n) .

מיון מהיר במקרה הטוב לוקח O(nlgn) אבל במקרה הטוב k = O(1) כלומר מספר קבוע כלשהו ולכן לא יהיה לנו מה למיין, ונקבל זמן ריצה במקרה הטוב של O(n) .

זמן ריצה במקרה הרע – select במקרה הרע תרוץ בזמן של O(n^2) (במקרה של חוסר מזל קיצוני כפי שמתואר בעמוד 155).

כמו כן במקרה ש k הוא לא מספר קטן קבוע ובמקרה של חלוקות גרועות נקבל זמן ריצה של O(k^2) (לפי עמוד 125) . סה"כ נקבל זמן ריצה במקרה הגרוע של O(n^2+k^2) .

הסבר לתוצאות: כיצד הניתוח התיאורטי מתיישב עם התוצאות האמפיריות?

כזכור התוצאות הראו ניצחון לאלגוריתם של הערימה עבור k גדול מספיק ואילו עבור k קטן התוצאות הראו ניצחון לאלגוריתם השני.

עבור k = n , נקבל זמן ריצה של O(nlgn) עבור שני האלגוריתמים.

כיצד נסביר שאלגוריתם הערימה מבצע כמעט מחצית מההשוואות שמבצע אלגוריתם הסלקט?

נשים לב לכך שניתוח הזמן עבור אלגוריתם הסלקט הוא עבור הזמן הממוצע, ואילו עבור אלגוריתם הערימה הוא עבור הזמן הגרוע. כמו

כן ניתוח הזמן של אלגוריתם הסלקט הוא בהנחה שהאיברים שונים זה מזה, כפי שראינו הנחה זו לא מתקיימת כאן ולכן יתכנו סטיות משמעותיות

לרעה בזמן הריצה של אלגוריתם הסלקט, ואילו באלגוריתם הערימה כמעט ולא יתכנו סטיות לרעה בשל החסם העליון על זמן הריצה.

הסברנו אם כך את הסיבה האפשרית לתוצאה הזאת.

עבור k קטן משמעותית מ – n קיבלנו תוצאות לטובה עבור אלגוריתם הסלקט. ניתן להסביר זאת בכך שהראנו שזמן הריצה של אלגוריתם זה תלוי

ב-k הרבה יותר מהאלגוריתם הערימה. למעשה, הראנו אף שהמקרה הטוב של אלגוריתם הסלקט הוא עבור k קטן יחסית לn ובמקרה הזה זמן הריצה יהיה לינארי.

אלגוריתם הערימה מושפע הרבה פחות מ-k יחסית לאלגוריתם הסלקט, למרות שגם מספר ההשוואות שהוא מבצע יקטן כאשר k יקטן וn לא ישתנה.

הרצה לדוגמה על סדרה בגודל 500 איברים:

