# קידוד ואלגוריתמים לזיכרונות 236379

## עינב הוברמן, בנימין וורניק ואלישבע רבינוביץ'

**הצגת הבעיה**

בהינתן מילים מאורך ו- מילים מאורך , יש למצוא אלגוריתם יעיל המחשב את הקב': .

כלומר, מציאת אלגוריתם המחשב את החיתוך של מס' כדורי הכנסה ומחיקה.

זהוי הרחבה של עבודה הנעשתה במאמר [The Intersection of Insertion and Deletion Balls](ieeexplore.ieee.org/document/9611515)

**הקדמה**

במאמר מוצג אלגוריתם יעיל עבור המקרה של . נקרא לאלגוריתם זה ID(x,y,n) כאשר x מאורך n-t ו-y מאורך n+t.

ראשית האלגוריתם מחשב את רשימת האינדקסים הימניים-ביותר במילה y אשר תת-המחרוזת של y הנוצרת מאינדקסים אלו נותנת את המילה x. חישוב זה הינו רקורסיבי ומצריך טבלת תכנון דינמי לחישוב ה-lcs של המילים x ו-y וטבלת התאמה של המילים (טבלה בינארית אשר יש בכניסה ה-ij את הערך 1 אם ורק אם ). לאחר מכן האלגוריתם מחשב עבור כל סט של אינדקסים אילו אינדקסים יש להוסיף מ-y כדי לקבל מחרוזת מאורך n.

**סיבוכיות:**

לפי המאמר, הוא חסם על זמן הריצה הריצה הממוצע של האלגוריתם.

נסמן ב- את הסיבוכיות של האלגוריתם. במקרים מסוימים ההקשר יהיה סיבוכיות ממוצעת ובהקשר אחרים סיבוכיות במקרה הגרוע (ההקשר יהיה מובן).

**אופן יצירת קלטים לטסטים**

בכל טסט הגרלנו ווקטור באורך n ויצרנו ממנו באופן רדנומלי את קבוצות המילים הארוכות והקצרות על מנת להבטיח שהחיתוך לא יהיה ריק. עשינו זאת על ידי הגרלת t אינדקסים להכנסת t ערכים אקראיים שערכם 0/1 על מנת ליצור את המילים הארוכות ועל ידי הגרלת t אינדקסים למחיקת ערכים ליצירת המילים הקצרות.

**קישור לעבודה בגיט**

<https://github.com/einavhu/intersection-of-multiple-insertion-and-deletion-balls.git>

**אלגוריתם 1 – פתרון נאיבי**

**פסאודו קוד**

קלט: ווקטורים כך שלכל מתקיים *ולכל מתקיים וגם .*

*פלט: .*

*האלגוריתם:*

1. *לכל נחשב:.*
2. *החזר .*

**הסבר**

בהינתן וקטור supersequences מאורך k המכיל מילים מאורך n+t ווקטור subsequences מאורך k המכיל מילים מאורך n-t, נחלק את המילים ל-k זוגות זרים כך שכל זוג מכיל מילה מווקטור ה-supersequences ומילה מווקטור ה-subsequences.

לאחר שקיבלנו k זוגות, נמצא את לכל באמצעות אלגוריתם ID.

לאחר שקיבלנו k קבוצות, נחזיר את החיתוך שלהן.

מציאת החיתוך של הקבוצות נעשה באמצעות מעבר יחיד על כל הקבוצות וספירת מספר המופעים של כל מחרוזת על פני כל הקבוצות. מחרוזת שמופיעה כמספר קבוצות הID תימצא בחיתוך. באופן זה מציאת החיתוך נעשית בזמן ליניארי בסכום גדלי הקבוצות

הערה: אנו מניחים כי שני וקטורי הקלט שלנו מסודרים בצורה אקראית ולכן באיטרציה ה-i ניקח את המילה ה-i מווקטור ה-supersequence ואת המילה ה-i מווקטור ה-subsequence, בהנחה שזה שקול ללקיחת k זוגות באופן אקראי.

אלגוריתם זה הוא הפתרון הנאיבי לבעיה, שנשתמש בו בתור baseline כדי להעריך את הביצועים של האלגוריתמים המופיעים בהמשך.

**ניתוח סיבוכיות**

1. מציאת ID של כל זוג: . נבצע זאת עבור k זוגות כך שסה"כ נקבל .
2. מציאת החיתוך של הIDים: ליניארי בסכום גדלי הIDים. במקרה הגרוע נקבל ID מסדר גודל של , כפי שראינו במאמר המדובר, ולכן סכום הגדלים הוא לכל היותר .
3. סה"כ: .

אם הוא התוחלת של אז בתוחלת סיבוכיות האלגוריתם תהיה כלומר עבור קבועים וקבוצת מחרוזות אקראיים, האלגוריתם צפוי להיות לינארי ב-.

**תוצאות**

A screenshot of a computer

Description automatically generated with low confidenceזמן ריצה ממוצע של אלגוריתם 1 כאשר מקבעים את ו-

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

Chart, line chart

Description automatically generatedזמן ריצה ממוצע של אלגוריתם 1 כאשר מקבעים את ו-

Chart, line chart, histogram

Description automatically generated

Chart

Description automatically generatedזמן ריצה ממוצע של אלגוריתם 1 כאשר מקבעים את ו-

Chart, line chart

Description automatically generated

Chart, line chart

Description automatically generated

*מסט הגרפים הראשון (כאשר ציר האיקס הוא ) ניתן לראות כי כאשר מקבעים את הגידול בזמן הריצה הוא לינארי כפי שצפינו. עם זאת, ניתן לראות בסט הגרפים השני (כאשר ציר האיקס הוא ) שההתנהגות של זמן הריצה הממוצע הוא אקספוננציאלי ב-, דבר אשר מסתדר עם כך שכאשר מקבעים את ו-, זמן הריצה בממוצע תלוי בגודל של . מוקדם יותר הסברנו כי במקרה הגרוע גודל החיתוך של זוג וקטורים הוא וחסם על המקרה הממוצע הוא , מה שמתיישב עם התוצאות.*

*מסט הגרפים האחרון (כאשר ציר האיקס הוא ) רואים הבדל בהתנהגות עבור -ים גדולים וקטנים. כאשר גדול, זמן הריצה קטן ככל ש- גדל. דבר זה נוגד את האינטואיציה הראשונית שלנו, אך נותן פתח למחקר המשך על ההתנהגות של הגודל הממוצע של החיתוך כפונקציה של ו-. אפשר להסביר זאת באמצעות התבוננות על הביטוי אשר הינו חסם עליון על גודל החיתוך. ביטוי זה הוא וכן . כלומר יש יחסי כוחות מנוגדים בין ו- באקספוננט שיכול להסביר את ההבדל שאנו רואים.*

**אלגוריתם 2**

**פסאודו קוד**

קלט: ווקטורים כך שלכל מתקיים *ולכל מתקיים וגם .*

*פלט: .*

האלגוריתם:

1. בחר.י ע"פ כלל החלטה כלשהו.
2. חשב.י את .
3. אתחל .
4. לכל :
   1. לכל : אם אינו supersequence של עבור לאיטרציה הבאה בשלב 4.
   2. לכל : אם אינו subsequence של עבור לאיטרציה הבאה בשלב 4.
   3. .
5. החזר.י את .

**הסבר**

בהינתן וקטורים מאורך ו- וקטורים מאורך , ניקח מתוכם זוג וקטורים באופן מסוים (ע"פ הגרסה שתפורט בהמשך) כאשר מאורך ו- מאורך נמצא את   
 באמצעות אלגוריתם ID.

לאחר מכן, נעבור על כל הווקטורים ב- ונבדוק עבור כל אחד מהם האם הוא נמצא ב-deletion ball של כל אחד מהווקטורים מאורך בקלט וגם נמצא ב-insertion ball של כל אחד מהווקטורים מאורך בקלט. אם כן, נשאיר אותו. אם לא, נוציא אותו מ-. נסמן את הקבוצה שהתקבלה בסוף ב-

הערה: עבור כל מילה ב- הבדיקה מתבצעת ב-. זאת מכיוון שניתן ב- לבדוק אם מילה אחת היא subsequence של מילה שניה וזה שקול לבדיקה האם המילה היא ב-deletion/insertion ball של מילה אחרת.

סה"כ נקבל שבסוף ריצת האלגוריתם מתקיים: ולכן נחזיר אותה.

**נכונות**

נראה כי בסוף ריצת האלגוריתם מתקיים ע"י הכלה דו כיוונית:

וגם לכל ו- *מתקיים הוא תת מילה של ו- הוא תת מילה של   
 וגם לכל ו- מתקיים כי וגם   
.*

**ניתוח סיבוכיות זמן**

נסמן ב- את זמן הריצה של כלל ההחלטה.

*.*

**היוריסטיקות לבחירת הזוג**

מניתוח סיבוכיות האלגוריתם ברור כי נרצה להקטין את כמה שניתן. על כן, נציג מספר היוריסטיקות שבחרנו שמטרתן לאפשר ניחוש של "זוג מוצלח" שהחיתוך שלו קטן. בהמשך נשווה בין ההיוריסטיקות מבחינת סיבוכיות (זמן ריצה) וביצועים (עד כמה החיתוך של הזוג שנבחר קטן).

סימונים:

* : אורך ה-alternating sequence המקסימלי ב-.
* : מספר ה-runs ב-.
* : קב' המילים הקצרות ().
* : קב' המילים הארוכות ().
* : וקטור ה- של . כלומר אם האורך של ה- ה- ב- הוא אז .

היוריסטיקות:

1. .
2. .
3. .
4. arg   
   הערה על 4: אם ו- אינם באותו אורך, נרפד את באפסים בסוף, ובנוסף, אם ו- לא מתחילים באותה אות נוסיף ל- 0 בתחילתו כדי להשוות את האורכים.

היוריסטיקה 1 (הסבר):

ההיגיון מאחורי היוריסטיקה זו הוא שכל זוג שהיא פולטת מחיל מגבלה על אופי המחרוזות בחיתוך: כל המחרוזות בחיתוך יצטרכו לקיים שמספר הruns בהם קטן או שווה למספר הruns במחרוזת הארוכה שנבחרה וגדול או שווה למספר הruns במחרוזת הקצרה שנבחרה. נסביר מדוע.

כאשר מבצעים deletion במילה, מס' ה-runs במילה לאחר המחיקות לא יכול לגדול. באופן דומה, כאשר מבצעים insertion במילה, מס' ה-runs במילה לאחר ההכנסות לא יכול לקטון.  
 . יתר על כן, כל מקיים: .  
לכן, נראה לנו שיש סבירות כי היוריסטיקה זו תפלוט בממוצע זוג בעל חיתוך קטן.

היוריסטיקות 2+3 (הסבר):

במאמר מראים כי עבור *שתי מילים ו- , מתקיים כי  
 תלוי באורכי ה* alternating sequences*המקסימליים במילים.*

*מכאן שעבור* t = 1, *מתוך הסתכלות על אורך ה* alternating sequence*המקסימלי במחרוזות, נוכל לשערך את גודל החיתוך שלהם.*

*ערכנו מספר נסיונות כדי לראות האם ואיך ניתן להרחיב קשר זה למקרה כללי ().*

*בחרנו להסתכל גם על ההפרש המינימלי וגם על ההפרש המקסימלי כדי לנסות להבין את ההתנהגות של גדלי החיתוכים.*

*היוריסטיקה 4 (הסבר):*

*נסתכל על ו- של מילים באורך ו- באורך אשר ה- אינו ריק. אם ו- באותו האורך, אזי בהכרח כל מילה ב- גם כן בעלת אותו אורך של וקטור , ויתר על כן סדר ה--ים של האפסים והאחדים הוא אותו הסדר. מכאן המרחק האוקלידי בין ו- הוא כנראה לא גדול מידי (המרחק יהיה לפחות ולכל היותר ). ככל שהמרחק האוקלידי קטן יותר המשמעות היא שמכל ב- מוחקים מספר קטן יותר של איברים כדי להגיע ל-, ולכן החיתוך אמור להיות גם קטן יותר.  
יחד עם זאת, מכיוון שלא ניתן להבטיח ש- ו- יהיו מאותו אורך היה צורך בריפוד של . החלטנו שרירותית לדחוף את הריפוד בסוף מכיוון שאין לנו דרך לדעת בצורה יעילה מאיפה נעלמו ה--ים. יתר על כן, זמן הריצה של היוריסטיקה זו הוא ריבוע ב- לעומת היוריסטיקה 1 שהוא לינארי ב-. לכן, להיוריסטיקה זו יהיה ערך אם גודל הקבוצה המתקבלת יהיה משמעותית קטן יותר מאשר גודל הקבוצה המתקבלת מהיוריסטיקה 1.  
בעיקרון, ניתן להשתמש בהיוריסטיקה זו גם כשובר שוויון בהיוריסטיקה 1 במידה ויש יותר מאפשרות אחת, אך כפי שנראה בהמשך אין בכך צורך שכן זמני הריצה לפי היוריסטיקה 1 הם נמוכים ביותר.*

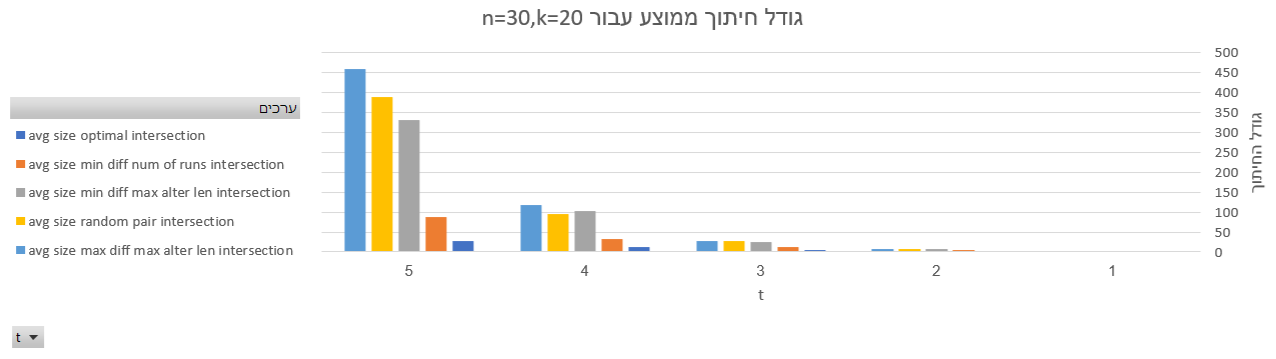
**ניתוח תוצאות ההיוריסטיקות והשוואתן**

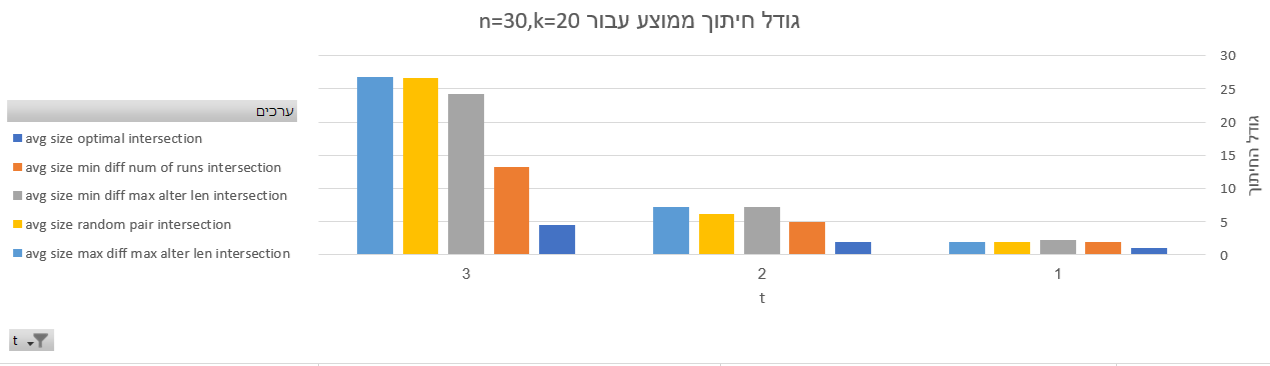
ניזכר כי השאיפה שלנו היא למצוא היוריסטיקה שמוצאת לנו את זוג הווקטורים שגודל ה-ID שלהם מינימלי (בכדי להקטין את זמן ריצת אלגוריתם 2, שתלוי בגודל הזה).

הרצנו את ההיוריסטיקות על קלטים אקראיים זהים ובדקנו את גודל החיתוך שהפלט של כל היוריסטיקה סיפק. ערכנו את הבדיקה על צירופים שונים של הפרמטרים כפי שיפורט להלן. נציג תוצאות המשוות בין גדלי החיתוך הממוצעים שהתקבלו מההיוריסטיקות, ננתח אותן ונסיק מהם מסקנות.

כדי שנוכל להעריך את טיב התוצאות, הצגנו בנוסף לפלט של ההיוריסטיקות גם שני baselines:

1. גודל קבוצת החיתוך המינימלית שיכולה להתקבל מבחירת זוג כלשהו (אותו מצאנו בעזרת חיפוש ממצה פשוט). תוצאה זו מוצגת בצבע כחול כהה בגרפים. השוואת יתר התוצאות למדד זה תראה לנו עד כמה אנחנו רחוקים מהאופטימום.
2. גודל קבוצת החיתוך המתקבלת מבחירה אקראית של מחרוזת קצרה ומחרוזת ארוכה באופן ב"ת זו בזו. בעזרתה נוכל לקבוע את הערך של ההיוריסטיקות: להיוריסטיקה יהיה ערך רק אם תפלוט קבוצה שהגודל שלה קטן מאשר הפלט של random.
3. **בינוניים**





תמונה שמכילה שולחן

התיאור נוצר באופן אוטומטי

מגרפים אלה עולה באופן ברור כי היוריסטיקה 2 נותנת את התוצאות הטובות ביותר, ואף גדֵלה בt בצורה מתונה הרבה יותר מיתר ההיוריסטיקות. למשל, הגדלת t מ1 ל5 הגדילה את גודל החיתוך הממוצע המתקבל מהיוריסטיקה 1 בערך פי 45, בעוד שהחיתוך המתקבל מההיוריסטיקה השנייה בטיבה (3) גדל כמעט פי 145.

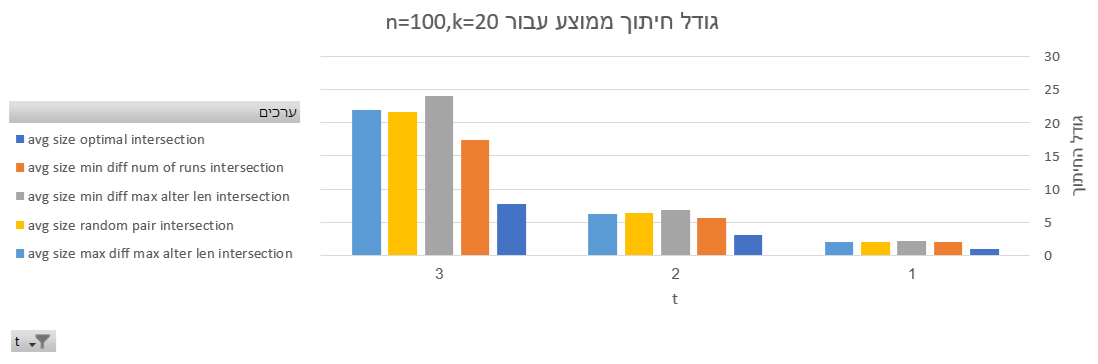
ניתן לראות שהיוריסטיקה 2 ו3 מספקות מידע "הפוך", כלומר, כאשר היוריסטיקה 2 נותנת חיתוך שגודלו גדול מעט מהחיתוך שמתקבל מהיוריסטיקת הבחירה האקראית, היוריסטיקה 3 נותנת חיתוך שגודלו מעט קטן יותר, וכן להיפך, ומכך שהתוצאות של שתי היוריסטיקות אלה מובילות לאותה מסקנה:

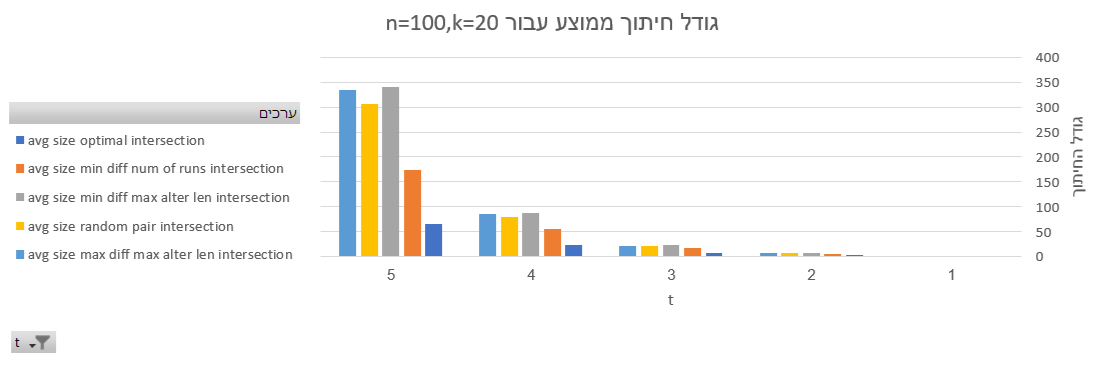
* עבור tים קטנים מאוד (, כדי להקטין את החיתוך, נרצה לקחת מחרוזות בעלות max alternating sequence באורך שונה ככל הניתן. תוצאה זו מתיישבת עם המאמר שהראה שכאשר, ID הוא הגדול ביותר עבור שתי מחרוזות שבהן המחרוזת הקצרה יכולה להתקבל מהארוכה באמצעות קיצור הalternating sequence המקסימלי, כלומר ההפרש בין אורכי הalternating sequence המקסימליים שלהם יהיה מירבי.
* עבור יתר ה tים, כדי להקטין את החיתוך, נרצה לקחת מחרוזות בעלות max alternating sequence באורך דומה.

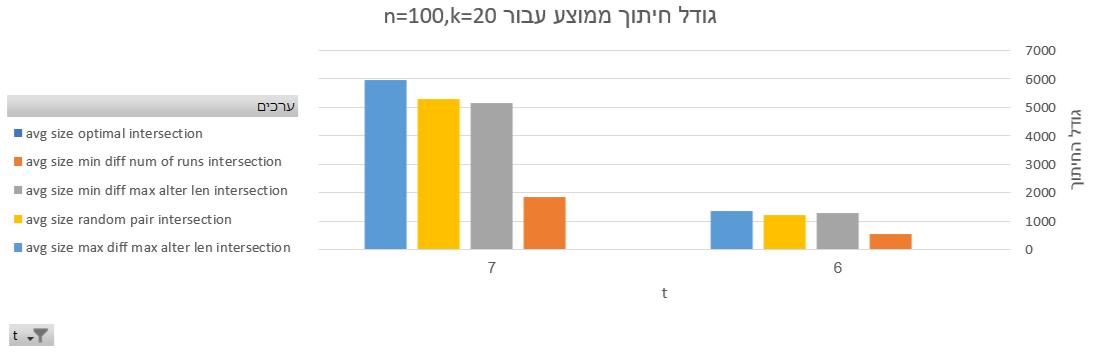
היות ושתי היוריסטיקות אלה נתנו תוצאות שונות רק במעט מהיוריסטיקת הבחירה האקראית, עבור כל הtים שבדקנו, נסיק שבמתכונת הנוכחית שלהם התרומה שלהם לאלגוריתם קטנה עד זניחה.

נציין כי ייתכן כי בשילוב עם היוריסטיקות נוספות (למשל בתור שובר שיוויון בין שתי בחירות שההיוריסטיקה רואה כשקולות) אולי ניתן להשתמש בהיוריסטיקות אלו כדי לשפר עוד את האלגוריתם.

1. **ההשפעה של הגדלת**

:





תמונה שמכילה שולחן

התיאור נוצר באופן אוטומטי

במקרה זה, כאשר הגדלנו את n פי 3.33, קיבלנו תוצאות דומות, אך עם כמה שינויים משמעותיים.

ראשית, הפער בין היוריסטיקה 1 ויתר ההיוריסטיקות קטֵן – למשל, בעוד שעבור t = 5 הגדלת n הגדילה את החיתוך שהתקבל מהיוריסטיקה 1 יותר מפי 2, החיתוך שהתקבל מהיוריסטיקת הבחירה האקראית *קטן* מעט, והחיתוך שהתקבל מהיוריסטיקה 3 קטן פי יותר מ1.35 (!).

בהמשך לכך, היחס בין היוריסטיקות 2 ו3 התהפך: כעת, גם עבור הtים הגדולים, היוריסטיקה 3 נותנת תוצאות טובות יותר. היוריסטיקה 2 כמעט ולא הושפעה מהשינוי בn.

סה"כ נוכל שוב להסיק שהיוריסטיקה 1 היא הטובה ביותר, אך רואים שיתר ההיוריסטיקות, ובכללן גם הבחירה האקראית, נותנות תוצאות משופרות. כדי להמשיך לבדוק את המגמה, השארנו את k כמו שהוא והגדלנו מאוד את n ((.



התוצאות מתאימות למגמה שראינו, ואף מחדדים את מה שקורה כאן: נראה שכאשר n מאוד גדול **כל** ההיוריסטיקות נותנות תוצאות מאוד דומות. בשלב הזה עוד יש יתרון קטן להיוריסטיקה 1, אך הוא הצטמצם מאוד. נוכל לנחש שכאשר ההיוריסטיקות כולן יתנהגו באופן זהה לבחירה רנדומית, ולכן במקרים בהם n מאוד גדול עדיף לוותר על ההיוריסטיקות שלנו ופשוט לבחור זוג אקראי. זה הגיוני כי

1. **ההשפעה של הגדלת**

כדי לבחון את ההשפעה של הגדלת , נשווה בין התוצאות עבור ו.

:



אנו מבחינים שבדומה למקרה של גדול, גם בגבול היוריסטיקה 2 שואפת לאלגוריתם הבחירה הרנדומי. לעומת זאת, היוריסטיקה 1 שומרת על היתרון היחסי שלה והגדלת k לא משפיעה עליה כלל! כמו כן נראה שהגדלת k משפיעה על היוריסטיקה 3 לרעה. הסבר?

הערה: להסביר למה לא התייחסנו לtים גדולים?

**סיכום התוצאות:**

|  |  |  |
| --- | --- | --- |
|  |  |  |
| היוריסטיקה 1 טובה משמעותית מבחירה אקראית ויתר ההיוריסטיקות (עבור כל הtים).  היוריסטיקות 3,4 נותנות עבור חלק מהtים ביצועים מעט טובים יותר מבחירה אקראית. | כל ההיוריסטיקות שואפות לתוצאה של בחירה אקראית. | אין השפעה על בחירה אקראית והיוריסטיקה 1  היוריסטיקה 2 שואפת לתוצאה של בחירה אקראית, והיוריסטיקה 3 פחות טובה מבחירה אקראית. (אפשר להשתמש בזה איכשהו?) |

**ניתוח תוצאות עבור אלגוריתם 2 עם היוריסטיקה 1 (הפרש מינימלי בין מספר ה-)**

Chart, line chart

Description automatically generatedזמן ריצה ממוצע של כאשר מקבעים את ו-

Chart, line chart

Description automatically generatedA picture containing text, sky, different

Description automatically generated

Chart, line chart

Description automatically generatedזמן ריצה ממוצע של כאשר מקבעים את ו-

Chart, line chart, histogram

Description automatically generated

Chart, line chart

Description automatically generatedזמן ריצה ממוצע של כאשר מקבעים את ו-

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

*מסט הגרפים הראשון (כאשר ציר האיקס הוא k ומקבעים את t,n) ניתן לראות שיפור בביצועים ככל ש-k גדל. מכיוון שההיוריסטיקה היא בחירת ההפרש המינימלי של מספר ה-, ככל שיש יותר מילים הזוג הנבחר יהיה בעל הפרש קטן יותר, מה שיקטין את הגודל של . ניתן לראות זאת גם בסט הגרפים השני (כאשר ציר האיקס הוא ) שעבור -ים גדולים יותר זמן הריצה אינו גדל אקספוננציאלית. למעשה, יש כאן טריידאוף בין ו-. כאשר מגדילים את , גודל החיתוך הממוצע של זוג וקטורים גדל, אך אם מסתכלים על גודל החיתוך הממוצע הנבחר* ***לפי ההיוריסטיקה הנ"ל*** *אז משחק תפקיד מאוד משמעותי וניתן לראות שאף משמעותי יותר מ- עצמו. לבסוף, מסט הגרפים האחרון ניתן לראות כי כאשר מגדילים את זמן הריצה גדל גם כן, וכאשר גדל זמן הריצה ב- גדל לינארית (עבור -ים קטנים ההתנהגות של זמן הריצה רועשת יותר). נשים לב כי הגודל של עדין משחק תפקיד גדול יותר מ- בזמן הריצה. ניתן לראות זאת בכך שנאלצנו להציג את הגרפים עבור בנפרד מאשר עבור שכן זמני הריצה הם בסדרי גודל שונים.*

***השוואת תוצאות של אלגוריתם 2 עם היוריסטיקה 1 ועם בחירת זוג רדנומלי***

*גרפים נבחרים עבור אלגוריתם 2 עם בחירה של זוג רנדומלי:*

*זמן ריצה ממוצע כאשר מקבעים את :*

Chart, line chart, histogram

Description automatically generated

Chart, line chart

Description automatically generated

נשים לב כי כאשר הזוג הנבחר הוא רנדומלי, כבר אין שיפורים משמעותיים בזמני הריצה עבור -ים גדולים יותר, ואנו חוזים שוב זמן ריצה שגדל אקספוננציאלית כאשר מגדילים את . זה כמובן מסתדר עם כל מה שהסברנו על גודל החיתוך של זוג וקטורים רנדומלי. יתר על כן, מכיוון שכל הקלט רנדומלי, כבר אינו משחק תפקיד משמעותי במקרה הממוצע. אנו כן עדים שוב לתופעה שזמני הריצה קטנים כאשר גדל בדומה לאלגוריתם 1.

**השוואת זמני ריצה של כל האלגוריתמים עבור (עבור יתר המקרים התוצאות דומות):**

הערה: בחרנו שלא להציג את זמני הריצה של אלגוריתם 2 עם יתר ההיוריסטיקות שכן ראינו כי היוריסטיקה 1 בעלת התוצאות הטובות ביותר, וכן מלבד בחירת זוג רנדומלי היא גם בעלת סיבוכיות זמן הריצה הנמוך ביותר.

זמני ריצה ממוצעים עבור קבוע ו- משתנה:

Chart, line chart

Description automatically generated

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated*זמני ריצה ממוצעים עבור קבוע ו- משתנה:*

*בבירור ניתן לראות כי ביצוע אלגוריתם 2 הם משמעותית טובים יותר מאשר של אלגוריתם 1, גם כאשר הזוג הנבחר באלגוריתם 2 הוא רנדומלי כפי שצפינו.*

***מסקנות סופיות וכיווני המשך:***

***להשלים***