ממ"ן 14

נדב אלון

322952631

**שאלה 1**

נחשב עלות מסלול מזערי עד הנקודה הi,j:

ניתן להגיע לנקודה הi,j רק מנקודות שמשמאל אליה (כלומר i-1) ובמרחק לא יותר מ1 אנכית (כלומר j-1 או j או j+1).

אז ניתן לרשום את הנוסחה כך:

נגדיר גם

אם i,j לא בתחום (גדולות מ n או קטנות מ1), ו.

נוכיח את נכונות הנוסחה:

יהי נקודה i,j.

אז ערך נוסחת הנסיגה הוא

נניח בשלילה שקיים מסלול עם ערך נמוך יותר שמגיע לנקודה i,j.

אז יש לו תת מסלול שנגמר בנקודה שמשמאלה ובמרחק 1 אנכית, משקלו w.

אז

אך

כי w הוא מסלול לאחד מהנקודות הנ"ל והביטוי משמאל הוא המינימום מהמכירים האופטימליים של נקודות מהקבוצה.

לכן

*ומהגדרה*

*ולכן*

*נתאר כעת אלגוריתם שמוצא את המסלול המזערי מהשכבה השמאלית לימנית:*

*ראשית, נגדיר מערך דו מימדי M בגודל n על n ונאתחל את הערכים ההתחלתיים*

*לאחר מכן, בשכבה הבאה, נאתחל את הערכיים כך:*

*כאשר דואגים לא לחרוג מגבולות המערך (כלומר לא משווים את).*

*איתחול כל איבר לוקח מספר קבוע של פעולות אלמנטריות (מציאת מינימום מ3 איברים קבועים, חיבור לערך קבוע).*

*נמלא את המערך, תהליך שיקח פעולות, כי לכל תא מבצעים מספר קבוע של פעולות ויש תאים.*

*לאחר שחישבנו את המערך שמלא בערכים , כל שנותר הוא למצוא את הערך המינימלי מהשכבה האחרונה (מציאת מינימום מn איברים = ), ומציאת המסלול המינימלי שהגיע אליו מבין ה3 שמסוגלים להגיע אליו.*

*כלומר לכל ערך M[i][j] שנקבל נשמור את הנקודה i,j ונחפש את הנקודה k,l כך ש.*

*יש 3 אפשרויות לנקודה הזאת אז לוקח זמן קבוע למצוא את הנקודה המתאימה.*

*כל פעם מתקדמים צעד אחד שמאלה ומבצעים מספר קבוע של פעולות (השוואות בין ערכים במערך וחיבור), לכן לוקח זמן לשחזר את המסלול המינימלי מהמערך שחישבנו קודם.*

*נכונות האלגוריתם נובעת מנכונות הנוסחה הרקורסיבית, והעובדה שאלגוריתם מחשב נכון את ערכי הנוסחה ומשחזר את הערכים שבנו את הערך הסופי מהמערך.*

*אז קיבלנו אלגוריתם לחישוב מסלול מזערי בזמן , מש"ל.*

***שאלה 2***

*קודם כל נשים לב שאם נמיין הפוך את רשימת התיבות לפי אורך או רוחב, ונסרוק את תיבות המגדל הגבוה ביותר, האינדקסים שלהם ילכו ויגדלו. זאת משום שאם מגדל יציב אז לתיבות הגבוהות יותר רוחב קטן שווה מלתיבות שמתחתיהן, ולכן יופיעו אחרי התיבות ברשימה ממוינת לפי רוחב.*

*כעת נתבונן בנוסחת הנסיגה הבאה:*

*כאשר OPT(i) הוא הגבוה המקסימלי של מגדל שמורכב מקבוצת התיבות עד i.*

*נוכיח את נכונות הנוסחה:*

*יהי i.*

*יש שני אפשרויות: התיבה i היא חלק מהמגדל הגדול ביותר, או שהיא לא.*

*אם i היא במגדל הגדול ביותר, אז התיבה שמתחתיה צריכה להיות מסוגלת לתמוך בה. ניתן להניח שכל הקופסאות שלפניה ברשימה הן בעלות רוחב גדול מi לפי המיון שבוצע בהתחלה, אז צריך להגביל את התיבה הבאה לתיבה עם אורך גדול מאורך תיבה i. אורך המגדל הזה מחושב ע"י הערך כאשר j היא תיבה עם אורך גדול משל i.*

*נניח בשלילה שקיים מגדל שמכיל את i וגבוה יותר מהערך הנ"ל. נתבונן בתיבה שמתחת לi. אורכה ורוחבה בוודאות גדולות משל i אחרת המגדל לא יציב. אז האינדקס שלה j קטן משל i, ואורכה גדול משל i, ולכן ערך המגדל שנגמר בה שגובהו H מקיים:*

*כי פונקציית הmax מכילה את ה-j המדובר בה, ולכן לא קיים מגדל שמכיל את j וגבוה מהערך הנ"ל.*

*אם i לא במגדל הגדול ביותר, אז המגדל הגדול ביותר עד i גובהו שווה למגדל הגדול ביותר עד i-1 באופן טריוויאלי.*

*לכן הנוסחא נכונה.*

*נתאר כעת את האלגוריתם:*

*קודם כל נמיין את הרשימה לפי רוחב, לוקח nlogn זמן ריצה.*

*אז ניצור מערך M בגודל n ונאתחל M[1]=h(1).*

*לאחר בלולאה שרצה על i מ2 עד n נחשב את הערך M[i] לפי ערכי המערך הקודמים כך*

*[יש לשים לב שאם אין ערך j שמקיים אז עדיין צריך להוסיף לh(i) את הערך 0].*

*לכל איטרציה של הלולאה מתבצע לכל היותר n השוואות (מציאת מינימום ממערך שגודלו חסום ע"י n לוקח זמן), לכן הזמן הכולל של הלולאה הוא .*

*כדי לשחזר מהמערך את המגדל הגבוה ביותר כל שעלינו לעשות הוא לחזור אחורה ולמצוא את הדרך שממנה הגענו אל M[n]. בכל צעד מתקדמים אחורה במערך לפחות צעד אחד לכן לכל היותר n איטרציות, ומשווים את הערך הנוכחי לn ערכים שונים לכל היותר, אז שחזור האינדקסים המשומשים יקח לכל היותר פעולות.*

*לכן זמן הריצה הכולל הוא .*

***שאלה 3***

*א.*

*לפי הגדרה, מתקיים*

*אם נציב בנוסחה נקבל:*

*נקבע את הערכים:*

*והנוסחה נכונה.*

*בנוסף, נציב ונקבל:*

*נקבע את הערכים:*

*והנוסחה נכונה.*

*הפולינומים r,s,q הם ממעלה 0 או 1, לכן ניתן לייצג אותם כך:*

*נציב a=c=1,*

*נקבל*

*בנוסף נתון*

*קיבלנו את הנוסחה הבאה:*

*ב.*

*בשביל לחשב פולינום אינטרפולציה מסדר n, צריך שני פולינומי אינטרפולציה מסדר n-1. הגיוני אם כן לחשב את כל פולינומי האינטרפולציה מהסדר הראשון, בעזרתם לחשב את כל הפולינומים מסדר 2, וכ"ו.*

*תיאור האלגוריתם:*

*ראשית נציב במערך M בגודל n (כאשר n=j-i) את ערכי הפולינום בנקודות.*

*אז נעשה לולאה על k מ1 לj-i כדי לחשב את הפולינומים מסדר k.*

*נעשה זאת כך:*

*לכל פולינום נחשב את מקדמיו בעזרת הפולינומים שחושבו באיטרציה הקודמת של הלולאה.*

*אם חיבור וחיסור פולינומים לוקח זמן ריצה כשm הוא הסדר הגדול יותר מביניהם, אז לכל חישוב של פולינום מסדר k לוקח זמן ריצה.*

*בכל איטרציה מחשבים לכל היותר n פולינומי אינטרפולציה, אז זמן ריצה של איטרציה אחת הוא .*

*מבצעים n איטרציות, לכן זמן הריצה הכולל של הלולאה הוא .*

*נכונות האלגוריתם מובטחת מנכונות הנוסחה הרקורסיבית.*

*יש לשים לב שניתן להשתמש בשני מערכים בגודל n כל אחד- אחד לשמור את הערכים הקודמים ואחר לאכסן ערכים חדשים, כי חישוב הפולינומים לא דורש פולינומים מסדר נמוך יותר מאחד פחות הסדר שלהם.*

*ג.*

*ראשית נציב במערך את הערכים ההתחלתיים*

*k=1:*

*k=2:*

*k=3:*

*k=4:*

*כנדרש.*

***שאלה 4***

*א.*

*האלגוריתם מחשב לכל צומת את המרחק המינימלי שלה מהצומת r.*

*נוכיח באינדוקציה על מספר הקשתות במסלול המינימלי בין השורש אל הצומת:*

*אם 0 אז המרחק 0, ואכן המערך מאותחל כ0.*

*נניח שלכל צומת עם n קשתות במסלול המינימלי שלה האלגוריתם מחזיר בA[v] את המרחק המינמלי שלה מr.*

*יהי v צומת עם n+1 קשתות במסלול המינימלי שלה מr, ויהי u הצומת הלפני אחרונה במסלול הקצר ביותר מr אל v.*

*נניח בשלילה שהמסלול הקצר ביותר אל u לוקח יותר מn קשתות. אז המסלול הקצר ביותר אל v היה לוקח יותר מn+1 קשתות, בסתירה לעובדה שיש n+1 קשתות במסלול המינימלי לv.*

*לכן ניתן להניח שמספר הקשתות במסלול אל u הוא n או פחות. לכן מהנחת האינדוקציה, יש איטרציה באלגוריתם שערך המערך A[u] משתנה לערך המסלול המינימלי אל u. באיטרציה הבאה של האלגוריתם הוא מוודא שבA[v] יש את ערך המסלול המינימלי (אם כבר מעודכן אז בכל מקרה היה, ואם לא מעודכן אז באיטרציה הזאת הוא מתעדכן).*

*לכן בוודאות כשהאלגוריתם יסיים, המערך יכיל את הערך הדרוש בו.*

*מש"ל.*

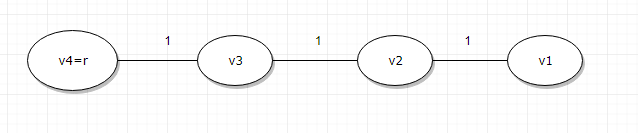
*ב.*

*מספר האיטרציות בלולאה החיצונית הוא מקסימום אם בכל איטרציה רק שכבה אחת מתעדכנת לאורך המסלול המינימלי. זה קורה אם הסדר הלקסיקוגרפי הוא מנוגד לסידור על פי מרחק מהשורש- כך בכל איטרציה מתעדכנת שכבה אחת אחרי שבודקים אם ניתן לעדכן את שאר השכבות שאחריה, בניגוד לסריקה על פי המרחק מהשורש שלוקח איטרציה חיצונית אחת.*

*במקרה זה מספר האיטרציות הוא מספר השכבות בסריקה לרוחב, כי בכל איטרציה מובטח לפחות ששכבה אחת תקבל את הערך הנכון שלה. מספר השכבות המקסימלי האפשרי הוא n-1 אם הגרף הוא חד מימדי, כלומר לשורש יש שכן אחד, ולכל צומת אחר יש לכל היותר שני שכנים, הגרף קשיר, והסדר הלקסיקוגרפי עובר מרחוק לקרוב.*

*סדרת גרפים שמקיימת את התנאי הזה היא סדרת הגרפים*

*דוגמא לגרף שכזה היא:*



*ג.*

*סדרת גרפים מתאימה לתנאי הזה היא שרוך (כדומה לסדרה הקודמת, ומכאן שמספר הצלעות שלהן שווה) שהסדר הלקסיקוגרפי שלה שונה: השורש הוא האחרון, השכן של השורש לפני אחרון, והשאר ממויינים בסדר המרחק מהשורש. כך האיטרציה הראשונה תעדכן את המרחק של השכן היחיד של השורש, והאיטרציה הבאה תאפשר עדכון של שאר הצמתות בגלל שלכל צומת מעודכנת אחרי הצומת האחרונה במסלול הקצר ביותר שלה, ולכן ניתן לעדכן גם אותה.*

*הגדרה ממשית לסדרת הגרפים:*