# שאלה 1

נתון סריג ריבועי בגודל עם מחירים אי שליליים לכל תא המסומן לפי הפונקציה . על האלגוריתם למצוא מסלול "זול" ביותר מצד שמאל של השריג (נסמן כאשר ) לצד ימין של השריג (נסמן כאשר ), כאשר התנועה האפשרית בשריג היא רק עבור השכנים הימניים של תא המקור.

## רעיון האלגוריתם:

נגדיר מטריצה בגודל (כלומר הוספנו 2 שורות) כאשר לכל בטווח מתקיים ש- הוא המחיר המינימלי מקודקוד כלשהו מצד שמאל ועד האיבר בשריג. בשתי השורות שהוספנו, שנגדיר שהן כאשר או הערך הוא . אנחנו נשתמש בשתי השורות האלה כדי לפשט את החישוב.

כלומר אם נתחיל לבנות נוסחה, נקבל:

* לכל עבור או על פי הגדרה נציב .
* לכל עבור (העמודה הכי שמאלית) נציב את מחיר האיבר עצמו: . זה כי מסלול המוביל מעמודה שמאלית לתא בעמודה זאת הוא בעל מחיר המבוסס רק על מחיר האיבר עצמו.
* עבור שאר הערכים, צריך לחשב. יהי תא כאשר (כלומר לא העמודה השמאלית). במסלול לתא זה מצד שמאל יכולים להשתתף רק התאים השכנים השמאליים שלו (לפי הגדרת התנועה בשריג), ולכן המסלול בעל המחיר הקטן ביותר הוא הסכום המינימלי בין מחיר האיבר ומחיר המינימלי לקודקודים השמאליים שלו, כלומר ערכם ב-, וכן קיבלנו שעבורו:  
  הערה: נשים לב שאם מדובר בשורה או אז עבור ה-"גלישה" עם ו- בהתאם נקבל ערך שהוא , כאשר בשני המצבים האחרים ב- נקבל מספר סופי, ולכן בתוצאת ה- תמיד נקבל מספר סופי.

כלומר, לפי הגדרת , נקבל שהמחיר המינימלי מצד שמאל לצד ימין הוא .

*כעט כדי לשחזר, נעשה "טיול" אחורה מהמחיר הזה. בתור התחלה נמצא עבור איזה ערך המחיר הזול מפסקה קודמת הוא נמצא – זהו הקודקוד האחרון במסלול המבוקש. בכל שלב בלולאה ניקח את המחיר הנוכחי והתא הנוכחי נמצא עבור איזה ערך מתקיים (אם יש כמה שמתאימים נבחר אחד מהם – אין הבדל). לפי ערך זה נתקדם בשחזור לתא ונחזור על הלולאה.*

## האלגוריתם:

עבור שריג בגודל ופונקציית מחיר :

1. נגדיר מטריצה בגודל .
2. לכל נגדיר וגם .
3. לכל נגדיר .
4. לכל בצע:
   1. לכל בצע:
5. נחשב . את ה- שעבורו היה הערך המינימלי נסמן ב-.
6. לכל ערך בצע:
   1. אם אז
   2. אם אז
   3. אם אז
7. לכל החזר את האיבר מהשריג בתור האיבר ה- במסלול.

## הוכחת נכונות:

בשלבים 1-3 אנחנו מאתחלים את המסגרת של מטריצה . בשלב 4 מתבצע השלב הרקורסיבי של חישוב . נשים לב שהלולאה החיצונית בשורה 4 היא מתחילה מ-2 (העמודה הראשונה כבר חושבה) ותמיד גדלה למעלה, ולכן עבור כל איטרציה , כל העמודות הערך המתאים ב- כבר חושב. ולכן עבור כל הערכים כבר חושבו ושמורים ב- ולכן הנוסחה תחת שורה מחזירה את הערך הנדרש לפי הרעיון האלגוריתמי. ולכן בסוף שלב 4, המטריצה מכילה את כל החישובים המתאימים לרעיון האלגוריתמי ואפשר להשתמש בערכים שלה להמשך.

בדומה לרעיון האלגוריתמי ולפי הגדרת אנחנו נקבל בשלב 5 שהמשתנה יהיה המחיר המינימלי מבין המסלולים מצד שמאל לצד ימין בשריג. לפי צורת החישוב קיים כך ש-. אם הוא לא יחיד, אז נבחר אחד מהם (כלומר יש מספרים מסלולים שהמחיר שלהם מינימלי – אבל האלגוריתם צריך להחזיר רק אחד, ולכן נבחר אחד מהם בצורה שרירותית). למעשה ערך הזה הוא הערך המתאים לאיבר האחרון במסלול הזול ביותר.

בשלב 6 אנחנו משחזרים את המסלול הזול ביותר. לפי צורת החישוב של , מתקיים שהערך של כל עבור איבר במסלול הוא סכום של מחיר האיבר עצמו מינימום ערכי של שכניו השמאליים, ולכן האיבר הקודם במסלול המתאים לעמודה הוא האיבר שערך שלו מתאים לערך המינימלי, שבדיקה זאת מתבצעת בשלב 6.

בשלב 7 אנחנו מחזירים בצורה יותר "קריאה" את המסלול כאוסף איברים.

## ניתוח סיבוכיות זמן:

עבור שלבים 1-3, נדרשים הצבות. בשלב 4 הרקורסיבי, יש לנו איטרציות כאשר בכל איטרציה החישוב בעל מספר קבוע של פעולות אריתמטיות בזמן קבוע, ולכן כל שלב 4 הוא בסיבוכיות בשלב 5 החישוב הוא בסיבוכיות . בשלב 6 יש איטרציות, כאשר בכל איטרציה יש מספר קבוע של פעולות אריתמטיות, ולכן גם פה . שלב 7 הוא גם בסיבוכיות . ולכן סה"כ הסיבוכיות היא:

# שאלה 2

האלגוריתם מקבל רשימה של תיבות מלבניות ופונקציות לכל המייצגות אורך, רוחב וגובה בהתאמה, כאשר נתון שכל האורכים, הרוחבים והגבהים שונים (כל קבוצה בנפרד). האלגוריתם מחזיר מגדל חוקי בעל גובה מקסימלי.

## רעיון האלגוריתם:

נגדיר בתור הגובה המרבי של מגדל חוקי כך שבראשו יש את תיבה . קיימים שני מצבים שונים:

* כל המגדל הוא למעשה התיבה לבדה, ולכן הגובה הוא .
* תיבה עומדת מעל מגדל חוקי אחר, שבראש מגדל זה נמצאת תיבה אחרת שנסמן ב-. נבחר את המגדל האופטימלי עבור .

בשביל לכסות את כל האפשרויות נחשב משותף ונקבל:

נשים לב שאם נמיין את כל התיבות לפי הרוחב בסדר יורד, אז כל חישוב של תלוי בערכים של כאשר , כלומר יש בסיס לתכנות דינאמי.

בזמן שאנחנו מחשבים את , נחשב גם את שהוא מייצג מה התיבה המתאימה שמעליה שמנו את תיבה , כאשר אם כל המגדל הוא התיבה עצמה (אין מישהו מתחת) אז נציב .

לבסוף התוצאה של האלגוריתם הוא מגדל מקסימלי שגובהו ואז נשאר פשוט לשחזר בחזרה באמצעות .

## האלגוריתם:

1. מיין את התיבות לפי רוחב התיבה בסדר יורד (כלומר אם"ם ).
2. נגדיר שני מבנים בגודל .
3. עבור בצע:
   1. וגם
   2. עבור בצע:
      1. אם מתקיים וגם אז:
4. נחשב . את ה- שעבורו היה הערך המקסימלי נסמן ב-.
5. עבור כל עוד מתקיים בצע:
6. הפוך את מבנה והחזר את התיבות המתאימות לאינדקסים.

## הוכחת נכונות:

בזכות העובדה שאין חזרות בערכי הרוחב של התיבות, המיון בשלב 1 הוא פשוט (אין צורך להסתבך עם חזרות).

בשלב 3 אנחנו מחשבים את . בזכות כך שמיינו את התיבות לפי סדר יורד, נקבל שמתקיים אם"ם . ולכן הנוסחה מהרעיון האלגוריתמי שווה ערך לנוסחה:

כלומר מספיק לבדוק רק את התיבות באינדקסים . אם במהלך הלולאה הפנימית לא נמצאו תיבות שגדולות יותר באורכן וגם נותנות גובה מגדל טוב יותר, אזי אנחנו לא משנים את הערך ההתחלתי שהצבנו בשלב , אשר מתאים למצב בו התיבה היא כל המגדל בפני עצמו. אם במהלך הלולאה נמצא תיבה מתאימה, אבל זו תיבה שאם נשים מעליה את התיבה נקבל מגדל גבוה יותר ממה שמצאנו עד כה, ולכן נעדכן את ערך ונעדכן את תיבת ה-"מקור" . בסוף שלב נקבל שחישבנו בדיוק את הערך של ושל . ולכן בסוף כל שלב 3 נקבל שחישבנו את כל מבנה ו-.

בשלב 4 אנחנו נמצא מהו המגדל הגבוה ביותר, ואז נתחיל משחזור שלו. בזכות נכונות החישוב (לפי הרעיון), אנחנו יכולים לשחזר בחזרה על ידי מעבר ל-"אב קדמון" בעזרת עד שנגיע לערך . נשים לב שלא יכול להיווצר לנו "מעגל" של תיבות, מכיוון שכל הרוחבים והאורכים שונים, ולכן בהכרח קיים סדר, ולכן בגלל הגדרת מגדל "חוקי" שום תיבה לא יכולה לחזור על עצמה במגדל, ולכן בהכרח קיימת תיבה המקיימת והיא תהיה תיבה האחרונה ב-"טיול" של השחזור.

סה"כ לאחר שלב 5 קיבלנו את מבנה שהוא כל התיבות אבל בסדר הפוך, ונשאר רק להפוך ולהחזיר (כדי שבסיס המגדל תהיה התיבה הראשונה במה שמוחזר).

## ניתוח זמן ריצה:

בזכות העובדה שהשוואות הן פעולה אלמנטריות, שלב 1, שלב המיון בא בסיבוכיות אם נבחר באלגוריתם מיון טוב. בשלב 3 יש לנו איטרציות עבור הלולאה החיצונית, וחסום כמספר הזה עבור הלולאה הפנימית. ולכן סה"כ יש איטרציות, כאשר בכל איטרציה אנחנו מבצעים מספר קבוע של פעולות אריתמטיות, ולכן סיבוכיות שלב 3 היא .

שלב 4 הוא חישוב לינארי של . לפי הוכחת הנכונות, כמות האיטרציות בשלב 5 הוא לכל היותר , כאשר בכל איטרציה יש מספר קבוע של פעולות אריתמטיות, ולכן שוב סיבוכיות של . גם ההיפוך בשלב 6 הוא בסיבוכיות , ולכן סה"כ הסיבוכיות היא:

# שאלה 3

## סעיף א

פולינום מדרגה 0 או 1 זהו פולינום שהוא מהצורה . ולכן נסמן:

מתוקף הסימון כאינטרפולציה של הנקודות עבור הצבה של ערכים המשותפים לשני הסימונים נקבל את אותו הערך:

כעט להתאים למשוואה הראשונה, נדרוש כדי לפשט ש-. כלומר . כמו כן נדרוש ש- ונקבל את הערך מתאים.

כעט להתאים למשוואה הראשונה, נדרוש כדי לפשט ש-. כלומר . כמו כן נדרוש ש- ונקבל את הערך מתאים.

נחבר את שניהם ונקבל שוויון ל-0, כלומר:

נציב ונקבל:

יש שני פתרונות או . הדרישה ש- אינה הגיונית (זה בסופו של דבר איזשהן נקודות עבורן יש לנו ערכים), ולכן רק מתקבל. נציב ב- ונקבל:

*לסיכום קיבלנו שעבור הפולינומים המתאימים לנוסחה הם:*

## סעיף ב

### הקדמה לאלגוריתם:

נגדיר להיות האינטרפולציה של הנקודות .

נשים לב שכאשר , כלומר יש נקודה בודדת אז האינטרפולציה היא פשוט הערך , כלומר:

בשאר המצבים, לפי סעיף א נקבל:

נמיר לצורת :

ונקבל שאינטרפולציה של הפולינום היא .

### האלגוריתם:

1. נגדיר מטריצה בגודל .
2. לכל בצע
3. לכל בצע:
   1. לכל בצע:
4. החזר

### הוכחת נכונות:

הפתרון מסתמך על נוסחת הנסיגה מסעיף א (עם ההמרה הקטנה לצורת ), אבל יש להוכיח שבכל חישוב בתוך שלב התלות הרקורסיבית כבר חושבה.

עבור חישוב של יש לנו בחישוב תלות בתוצאות ו-. נשים לב שבאיטרציות אנחנו שומרים על קיום , ולכן יכולים להיות 2 מצבים:

* אם אז במצב זה שהוא חושב כבר בשלב 2.
* אחרת, נשים לב שעבור , החישוב חושב באיטרציה הקודמת של הלולאה החיצונית, ועבור החישוב בוצע באיטרציה הקודמת של הלולאה הפנימית (נשים לב שהלולאה הפנימית היא בסדר יורד).

*ולכן כל חישוב בשלב תואם לחישוב מנוסחת הנסיגה, ולכן מעצם נכונותה אנחנו מקבלים שהאינטרפולציה של היא .*

### ניתוח זמן ריצה:

בשלב 2, יש לנו איטרציות שבכל אחת אנחנו מציבים ערך מסוים, כלומר . בשלב 3 יש לנו איטרציות עבור הלולאה החיצונית, וחסום כמספר הזה עבור הלולאה הפנימית. ולכן סה"כ יש איטרציות. בחישוב של שלב אנחנו פעמיים מכפילים פולינום מסדר קטן ב-1 מהנוכחי בפולינום מסדר 1 ומחלקים בקבוע, כלומר יש לנו סדר גודל של פעולות אריתמטיות בכל איטרציה! ולכן סיבוכיות האלגוריתם סה"כ היא .

## סעיף ג

עבור ו- נציב את הערכים ונקבל:

לפי שלב 2 אנחנו מחשבים את ונקבל:

ואז לפי שלב 3, כאשר :

כאשר :

כאשר :

כאשר :

ואכן מתקיים ש- כנדרש.

# שאלה 4

נסמן ב- את משקל המסלול המינימלי מ- ל- בגרף הנתון.

## סעיף א

**חישוב האלגוריתם:** בסוף הלולאה החיצונית, האלגוריתם מחשב לכל צומת כך שנקבל .

יהי צומת הנמצא ברכיב הקשירות של . כלומר קיים מסלול מ- אל . מכיוון ומשקלי כל הצלעות הם אי-שליליים, אז כל מסלול בעל משקל מינימלי הוא מסלול פשוט. ולכן האורך המרבי של כל מסלול פשוט היוצא מ- אל הוא (אם כל הצמתים מופיעים במסלול), ולכן לפי טענת העזר בסיום האיטרציה ה- הערך של הוא המינימלי מבין כל המסלולים מ- אל , ובפרט עבור המסלול בעל המשקל המינימלי, ולכן מתקיים . ואז באיטרציה ה- אין עדכונים (כי כבר הצבנו את המינימלי לכל הצמתים) והאלגוריתם מסיים את עבודתו.

עבור הצמתים שלא נמצאים ברכיב הקשירות של , כלומר לא קיים מסלול מ- אליהם, נניח בשלילה שהאלגוריתם טעה וסימן עבורם ערך שונה מ-. נבחר את הצומת הראשון שלא ברכיב הקשירות והאלגוריתם טעה בו. עדכון זה יכול לקרות רק אם קיים צומת כך ש- וגם . מכיוון ובחרנו את להיות הצומת הראשון שלא ברכיב הקשירות של אז ברכיב הקשירות של , ולכן קיים מסלול מ- (דרך ) אל . וזה בסתירה שהוא אינו ברכיב הקשירות שלו. ולכן עבור כל הצמתים שלא ברכיב הקשירות הערך הוא .

ובכך הוכחנו כי בסיום האלגוריתם מתקיים .

**טענת עזר:** בסיום האיטרציה ה--ית של הלולאה החיצונית, לכל צומת כך שקיים מסלול מ- ל- באורך לכל היותר מתקיים יכיל את המשקל המינימלי של כל המסלולים הללו.

נוכיח באינדוקציה:

בסיס האינדוקציה: עבור , הצומת היחידי שמקיים שהוא נמצא במרחק 0 מצומת הוא הצומת עצמו שבאתחול מתקיים .

צעד האינדוקציה: נניח שהטענה מתקיימת עבור האיטרציה , נוכיח שהיא מתקיימת עבור האיטרציה ה--ית

יהי צומת המקיים . יהי מסלול בעל המשקל המינימלי מ- אל באורך מרבי . מכיוון ואנחנו לא במקרה , ולכן מתקיים כי קיים צומת המקיים וגם (כלומר צומת זה הצומת לפני האחרון במסלול ), כאשר זה תת המסלול ב-.

המשקל של מסלול הוא המינימלי מבין כל המסלולים מ- אל באורך מרבי . הוא באורך כי הורדנו צלע בודדת, והוא במשקל מינימלי בגלל שאם נניח בשלילה שקיים מסלול בעל משקל קטן יותר ובאורך מרבי , אז בשרשור עם הצלע היינו מקבלים מסלול באורך מרבי ומשקל קטן ממשקלו של וזה בסתירה להיותו מסלול בעל משקל מינימלי.  
אבל מכיוון ותת מסלול הוא באורך מרבי , אז לפי צעד האינדוקציה מתקיים יכיל את משקלו.

ולכן במהלך המעבר בלולאה הפנימית אנחנו נמצא את הרשת ולכן מתקיים .

*נניח בדרך השלילה כי . לפי החישוב והעדכון, משמעו שקיים צומת כך שקיימת הצלע וגם , כלומר קיים מסלול באורך מרבי כך שצומת זה הצומת לפני האחרון שלו. תת המסלול שלו הוא באורך מרבי ולכן לפי צעד האינדוקציה הוא לכל היותר משקלו של . ולכן הוא למעשה משקלו של המסלול*

*סה"כ קיבלנו שמתקיים .*

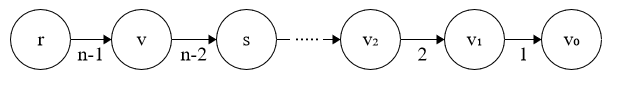
***מש"ל***

## סעיף ב

יהי גרף גרף בעל צמתים. לפי מה שהוכחתי בסעיף א מספר האיטרציות חסום במספר הצמתים, כלומר אם נסמן ב- את המספר האיטרציות המרבי, מתקיים .

על מנת שיתבצעו איטרציות, אנחנו צריכים שלא קיים ערך כך שלכל צומת אז (כלומר לכל ערך יש לפחות צומת אחד שהוא במרחק כזה מ-).

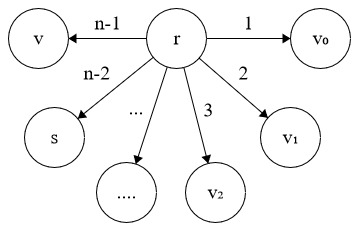
יהי ערך שהוא מספר הצמתים בגרף. נגדיר את הגרף להיות גרף "שרוך", כלומר לכל צומת פרט ל- יש רק קשת אחת שנכנסת אליו. כלומר . נגדיר שכל הקשתות מסודרות בסדר לקסיקוגרפי הפוך. כלומר, הוא נראה משהו כמו:

**

*רק צומת בודד מתעדכן ב- האיטרציות הראשונות (הצומת האחרון שלא עודכן באיטרציה הקודמת – בזכות הסדר* לקסיקוגרפי *הפוך לקשתות) ובאיטרציה ה--ית לא מתבצע עדכון ויוצאים מהאלגוריתם.*

## סעיף ג

נגדיר סדרת גרפים שעבור כל מתקיים . קל לראות שמתקיים מהגדרה זאת ש- כנדרש.



האורך המרבי של מסלול בגרף זה הוא 1, ולכן לפי טענת העזר מסעיף א בסוף האיטרציה הראשונה כל ערכי הוגדרו לכל צומת . ולכן באיטרציה השנייה לא מתבצע שום עדכון ויוצאים מהאלגוריתם.