**אלגוריתמים**

**ממ"ן 41 - תשובות**

מספר קורס: 20417

מנחה: אורן רות

שם הסטודנטית: ברית בן-דוד

ת"ז: 204879365

היי אורן, לאחר עבודה מאומצת אני בוחרת להגיש ממ"ן חלקי ולא לכתוב תשובות חלקיות שלא הבנתי לעומק או להעתיק. מגישה את שאלות 1, 2 וחלק קטן משאלה 3.

(אם אותן אתה בודק זכיתי, אם לא אז לא נורא – לפחות ניסיתי).

**שאלה 1:**

**מסלולים מזעריים בשריג עם מחירים על הקודקודים.**

מטרה:

הצגת אלגוריתם למציאת מסלול במחיר מזערי מהשכבה השמאלית ביותר לימנית ביותר, כשמחיר מסלול מוגדר כסכום מחירי הנקודות במסלול.

על האלגוריתם לבצע פעולות אלמנטריות בלבד (חיבור, חיסור, השוואה).

האלגוריתם:

נשתמש באלגוריתם תכנון דינאמי - נגדיר את תתי הבעיות ואת OPT:

נסמן ב-OPT(i,j) () את המסלול המינימלי החדש (הפתרון האופטימלי) של המסלול עד לאיבר הנוכחי (i,j).

**טענה:**

**הוכחת נכונות:**

בכל שלב, תת-בעיה, נביט במסלול האופטימלי עד לנקודה אליה הגענו – (i,j).

נדגיש כי קיימות 3 דרכים להגיע לנקודה (i,j):

* משמאל – מהנקודה (i-1,j).
* מאלכסון שמאלי עליון (שמאל למעלה) – מהנקודה (i-1,j+1).
* מאלכסון שמאלי תחתון (שמאל למטה) – מהנקודה (i-1,j-1).

שלבי חישוב המסלול המינימלי של תת-הבעיה:

1. בשלב הראשוני של חישוב המסלול נבדוק מה היה חישוב המסלול בכל אחת מ-3 הנקודות שציינו, ז"א . במידה והתוצאה היא 0, ז"א שלא היה קיים מסלול כזה עד כה, לכן נציב את הערך אינסוף במקרה כזה, על-מנת שהוא לא יבחר כמסלול המינימלי (כי הוא לא קיים). במידה והתוצאה שונה מאפס (אחרת), נשאיר את התוצאה כמו שהייתה.
2. כעת נבדוק מהו המסלול המינימלי עד ל- (i,j). מכל אחת מ-3 הדרכים להגיע לנקודה זו. לאחר-שמצאנו את הדרך המינמלית, נוסיף לה את מחיר הנקודה הנוכחית c(i,j) וזהו המסלול האופטימלי שקיבלנו עד כה.

כמובן שנשמור ערך זה על-מנת לעשות בו שימוש חוזר.

את המסלולים האופטימיליים שמצאנו בכל תת-בעיה נשמור במערך עזר M על-מנת שנוכל להשתמש בהם בהמשך.

ניתוח זמן ריצה:

מכיוון שמדובר בשריג ריבועי מסדר nxn (כאשר ) , נדרש לבצע חישוב עבור nxn תת-בעיות שכל אחת מהן היא המסלול האופטימלי עד לנק' מסוימת (i,j).

נוסחת הנסיגה בכל שלב היא , כאשר הוספת המחיר הנוכחי c(i,j) שקול לזמן ריצה של וסה"כ נקבל פעולות נדרשות בכל שלב. לכן עבור כל n השלבים ידרשו פעולות אלמנטריות, כנדרש.

**שאלה 2:**

**בניית מגדל יציב מתיבות**.

מטרה:

הצגת אלגוריתם תכנון דינאמי לבניית מגדל יציב בגובה מירבי, בהתאם לדרישות השאלה.

האלגוריתם נדרש לרוץ בזמן (פעולות חיבור, חיסור והשוואה של מימדים l(i),w(i),h(i) נחשבות פעולות אלמנטריות שמתבצעות בזמן ).

האלגוריתם:

ראשית, נסמן ב-OPT(k) () את הגובה המקסימלי אליו ניתן להגיע בבניית מגדל יציב, כנדרש בשאלה.

1. נגדיר משתנה עזר – maxh ונאתחל אותו במשתנה 0. משתנה זה ישמור בכל שלב נתון את הגובה המקסימלי אליו ניתן להגיע באמצעות בניית מגדל מתיבות מסוימות.
2. נגדיר מערך עזר דו-מימדי M ובו nX2 תאים.

* בתאים בשורה הראשונה M[0][k] () – ישמרו התיבות.
* בתאים בשורה השנייה M[1][k] () – יישמרו ערכים מספריים המייצגים את הגובה המקסימלי של המגדל שניתן להגיע אליו כאשר התיבה במיקום זה מהווה בסיס. נאתחל ערכים אלו בערך 0.

1. נמיין (מיון מיזוג) את n התיבות המלבניות ונשמור כל תיבה בתא M[0][k] במערך ((, לפי גודל שטח התיבה, לפי הנוסחה המוכרת: .

התיבה בעלת השטח הקטן ביותר תשמר בתא M[0], בעוד שהתיבה בעלת השטח הגדול הקצר ביותר תשמר בתא M[n-1].

1. כעת, עבור כל M[0][k] כאשר ועבור כל M[0][q] כאשר נבדוק את התנאים הבא נשתמש בנוסחה הבאה לבדיקה המסלול האופטימלי:
   * אם וגם (הרוחב והאורך של התיבות גדולים/קטנים, בהתאמה – המשמעות היא שניתן להניח את תיבה k על תיבה q).

וגם - אם (הגובה של k בתוספת הגובה המקסימלי שהגיע עד אליו גדול מהגובה השמור ב-k).

* + - אז -

ז"א, נבחר את הגובה הגבוהה יותר שיהווה כגובה האופטימלי – אם זה גובה התיבה לבדה או אם זה גובה התיבה בתוספת תיבות נוספות שנמצאות במגדל וחישבנו לפניה.

1. בסוף, נעבור על כל M[1][k] כאשר ונבדוק עבור כל הגבהים האופטימליים שמצאנו (כל גובה רלוונטי לתיבה אחרת), ונמצא מיהו הגובה המקסימלי ביותר. זו התוצאה שנבחר.

נסמן ונחזיר את התוצאה הזאת.

הוכחת נכונות:

בכל שלב, תת-בעיה, נביט בגובה האופטימלי עד לתיבה אליה הגענו.

נדגיש כי קיימות 2 דרכים להגיע לגובה האופטימלי (שהוא הגובה המקסימלי):

* העמדת התיבה לבדה – .
* העמדת התיבה מעל מגדל יציב שמצאנו עד כה, עד לתיבה q כלשהי - .

לפי 2 דרכים אלו ברור שעבור כל תיבה נמצא מהו הגובה המקסימלי שניתן להגיע אליו במגדל המורכב מתיבה זו (בצורה כלשהי).

לבסוף, נעבור על כלל הגבהים שמצאנו, עבור כל התיבות, ונמצא את הערך המקסימלי ביותר. ערך זה הוא הגובה המקסימלי של מגדל יציב שניתן להגיע אליו מהתיבות הנתונות.

ניתוח זמן ריצה:

ננתח כל שלב של האלגוריתם ולבסוף נבדוק את הסיבוכיות המשותפת:

1. – הגדרת משתנה עזר.
2. – הגדרת מערך עזר דו-ממדי.
3. – מיון מיזוג.
4. – מעבר עבור כל תיבה על כל התיבות שהשטח שלהן גדול ממנה והיא יכולה להיות מעליהן.
5. – מעבר על המערך ומציאת הגובה המקסימלי מהגבהים ששמרנו.

בסה"כ נבצע פעולות אלמנטריות, כפי שנדרש.

**שאלה 3:**

**אינטרפולציה באמצעות תכנון דינאמי.**

1. מטרה:

לכל נסמן ב- את פולינום האינטרפולציה של הנקודות .

נמצא 3 פולינומים פשוטים q(x), r(x), s(x) מדרגה 0 או 1, עבורם מתקיים:

*פתרון:*

אינטרפולציה היא שיטה בתחום האנליזה הנומרית, בפועל שיטה זו מסייעת לנו להעריך בקירוב תוצאה של פונקציה שאינה ידועה, ע"י נקודות במישור. ניתן לבצע פתרון לבעית אינטרפולציה באמצעות שיטת נוויל.

שיטת נוויל אינה מחברת מספרים מסדרי גודל שונים, כיוון שזה פוגע בדיוק של החישוב, לכן השיטה מבצעת חיבור של הפולינום בשלבים, כאשר בכל שלב ניתן לבצע חישובים על מספרים בסדרי גודל דומים (מאפשרת ליצור עץ חישוב).

נראה כי, לפי הגדרות האינטרפולציה, מתקיים:

נרצה להראות כי מתקיים גם \*\*:

נציב את ההגדרות ונבדוק כיצד ניתן לקבל את מה שאנחנו מחפשים:

ניתן לראות שכדי לקבל את \*\* צריך להתקיים התנאי הבא:

וברור כי גם:

כעת, נבנה את 3 הפולינומים הללו בצורה שיקיימו את התנאי ובהתאם לשיטת נוויל:

נציב את הפולינומים שמצאנו ונציב שוב את ההגדרות ונראה כי אנו מקבלים את מה שרצינו:

כנדרש.

לכן הפולינומים שמצאנו מתאימים.

1. מטרה:

בהינתן n נקודות נציג אלגוריתם תכנון דינאמי לבעיית האינטרפולציה, המבוסס על נוסחת הנסיגה מסעיף (א). לשם פשטות, נחשב פעולות אריתמטיות על מספרים כפעולות אלמנטריות.

אלגוריתם:

נסמן ב-OPT(I,j) את פולינום האינטרפולציה ().

לכן, לפי נוסחת הנסיגה שמצאנו בסעיף (א) מתקיים - \* :

ראיתי את התשובה אבל אני לא מעוניינת להעתיק ולא הצלחתי להבין לגמרי איך הגיעו לכך שצריך להתקיים:

אחרי שמגדירים זאת ברור לי למה שהנוסחה שציינתי ב-\* תתקיים, אבל כאן אני תקועה.

**שאלה 4:**

**כפל מטריצות ריבועיות (Strassen)**.

**סעיף ד':**

**טענה:** *מספר הפעולות האלמנטריות של האלגוריתם של Strassen הוא*  בלבד.

**הוכחה:**

*נזכיר כי באלגוריתם הוגדרו 7 המטריצות הבאות:*

בנוסף, הוגדרו הנוסחאות הבאות:

*נחשב את מספר הפעולות האלמנטריות הנדרשות לביצוע בעת הפעלת האלגוריתם ולאחר-מכן נשתמש בנוסחת הנסיגה לחישוב זמן הריצה שלו.*

1. *בסעיף ב' ווידאנו כי חישוב המטריצות כרוך ב-*7 פעולות כפל בלבד של מטריצות מסדר (כיוון שבכל P יש פעולת כפל אחת)

🡨 ז"א

1. נסכום את מספר פעולות החיבור/חיסור של המטריצות ב-  *ונראה כי נדרשות 10* פעולות כאלו. בנוסף, נסכום את מספר פעולות החיבור/חיסור של הנוסחאות שהגדרנו s,t,r,u ונראה כי קיימות שם 8 פעולות כאלו.

לכן, סה"כ קיימות באלגוריתם 18 פעולות חיבור/חיסור שנעשות בזמן של גודל המטריצה בריבוע

🡨 ז"א

לפיכך, נקבל שנוסחת הנסיגה לחישוב האלגוריתם היא:

*כעת, נשתמש בשיטת האב לפתרון נוסחת הנסיגה ונראה כי:*

***עבור מתקיים:***

*ולכן:*

***עבור מתקיים:***

ניתן לראות כי מתקיים מקרה 1 של שיטת האב:

קיים קבוע כך ש:

ולכן:

*הראנו כי מספר הפעולות האלמנטריות של האלגוריתם הוא , כנדרש.*