רקורסיה

מבוא למדעי המחשב - 67101

מרצה: אריה שלזינגר

מתרגלים: יפעת חדד ואורי מאיר

סוכם ע"י: שריה אנסבכר

סמסטר ב' תשפ"ג, האונ' העברית

אשמח לקבל הערות והארות על הסיכומים על מנת לשפרם בעתיד,
כל הערה ולו הפעוטה ביותר (אפילו פסיק שאינו במקום או רווח מיותר) תתקבל בברכה;
אתם מוזמנים לכתוב לי לתיבת הדוא"ל: sraya.ansbacher@mail.huji.ac.il.

לסיכומים נוספים היכנסו לאתר:
אקסיומת השלמות - סיכומי הרצאות במתמטיקה
https://srayaa.wixsite.com/math

1 מבנה

המבנה הכללי של פונקציה רקורסיבית הוא כדלהלן:

אלגוריתם 1 פונקציה רקורסיבית

הקלט יכול להיות כל דבר.

- אם הקלט מקיים את אחד מתנאי ההתחלה (ייתכן שיהיה יותר מתנאי התחלה אחד):
  - החזר ערך שנקבע מראש עבור תנאי זה.
- אחרת:
  - בצע פעולה כלשהי על מספר קריאות של הפונקציה הרקורסיבית עם קלטים "קרובים" יותר לאחד מתנאי ההתחלה (בנפרד).

♣ נשים לב שהקלט מוכרח להיות אחד מתנאי ההתחלה או כזה שקיימים עבורו מספיק קלטים קרובים יותר לתנאי ההתחלה שעוזרים במתן תשובה עבור קלט זה.

♣ ניתן להמיר כל פונקציה רקורסיבית לפונקציה המשתמשת בלולאות ולהפך, החיסרון העיקרי של פונקציות רקורסיביות ביחס ללולאות הוא כמות הזיכרון הנדרשת להפעלתן (הסיבוכיות זהה) שכן כל שלב של הרקורסיה דורש מקום בזיכרון כל עוד להא הגענו למקרה הבסיס, יתרונן של פונקציות רקורסיביות הוא שלעיתים קל יותר לכתוב, לקרוא ולהבין את הקוד בצורה רקורסיבית.

♣ לפעמים הקלט משמש בכמה תפקידים עבור הפונקציה ואם עבור חלקם צריך להכניס בכל שלב של הרקורסיה ערך שונה בעוד שבאחרים יש להכניס ערך קבוע[1] אז אי אפשר לבצע את הפונקציה בצורה רקורסיבית, במקרים כאלה יש ליצור פונקציית עזר רקורסיבית המתייחסת לכל היבט של הקלט בנפרד ובפונקציה העיקרית לבצע רק את הפיצול של הקלט לתפקידים השונים ולהכניס אותם לפונקציית העזר[2].

♣ יותר מזה אין לי מה להגיד מלבד העובדה שכמו תמיד דוגמאות עוזרות להבנת הנושא.

1.1 דוגמאות

דוגמה 1.1. עצרת ($n!$)

אלגוריתם 2 עצרת

הקלט הוא מספר טבעי $n \in \mathbb{N}_0$ (כולל 0).

- אם $n = 0$:
  - החזר 1.
- אחרת:
  - החזר את n כפול $(n-1)!$ (כלומר n כפול הפלט עבור $n-1$).

[1] דוגמה: אנחנו רוצים לבדוק אם $1 < x \in \mathbb{N}$ הוא חזקה של $1 < b \in \mathbb{N}$, בכל שלב נכפיל את b בעצמו ונבדוק אם הגענו ל-x או עברנו אותו, b משמש גם בתור הבסיס של הסדרה ההנדסית וגם המנה שלה ולכן אם נעשה זאת בצורה רקורסיבית נעבור רק על החזקות $b, b^2, b^4, b^8, \dots b^{2^n}, \dots$ כך שנוכל לעבור את x אפילו אם הוא באמת חזקה של b.

[2] במקרה של הדוגמה שבהערה הקודמת נבנה פונקציית עזר המקבלת שלושה קלטים שהם הבסיס, המנה והיעד (x) ובודקת אם הבסיס גדול מ-x או שווה לו ואם לא אז נכפול את הבסיס במנה ונבצע את הפונקציה שוב, בפונקציה העיקרית נכניס את b לפונקציית העזר גם בתור הבסיס וגם בתור המנה.

דוגמה 1.2. **סדרת פיבונאצ'י**[3]

אלגוריתם 3 סדרת פיבונאצ'י

הקלט הוא מספר טבעי $n \in \mathbb{N}_0$ (כולל 0).

- אם $n = 0$:
  - החזר 0.
- אם $n = 1$:
  - החזר 1.
- אחרת:
  - החזר את הסכום של מספר פיבונאצ'י ה-$n-1$ ומספר פיבונאצ'י ה-$n-2$ (כלומר את הסכום של הפלטים עבור $n-1$ ו-$n-2$).

2 גישוש נסוג (Backtracking)

דוגמה 2.1. איך פותרים מבוך? נתקדם מנקודת ההתחלה לצומת הקרובה (אם אין צומת קרובה אז יצאנו מהמבוך או שאין לו פתרון), נבחר אחת מהאפשרויות בצומת ונחזור על אותה הפעולה כאשר אם הצלחנו לצאת מן המבוך נחזיר שיש למבוך פתרון ואם נתקענו אז נחזור אחורה ונבחר באפשרות אחרת; אם נגמרו האפשרויות נחזור עוד אחורה לצומת שלפניה ואם הגענו לנקודת ההתחלה המקורית נחזיר שאין למבוך פתרון.

בדוגמה זו ראינו מקרה קלאסי של פתרון בעיה באמצעות "גישוש נסוג" (Backtracking), זוהי שיטה רקורסיבית לפתרון בעיות שעבורן ייתכן שיש לרקורסיה יותר מענף אחד: בדוגמה של המבוך בכל צומת ייתכן שיש יותר מפנייה אחת שאפשר לקחת בעוד שבפונקציית העצרת יש רק "פנייה" אחת שבה אפשר לבחור, גם בפונקציה של סדרת פיבונאצ'י יש רק דרך אחת לבחור בה בגלל שאנחנו מוכרחים לחשב את שני האיברים הקודמים בסדרה ואין לנו את היכולת לבחור לחשב רק אחד מהם.
לגישוש נסוג ישנן שתי גרסאות: באחת אנחנו רוצים לקבל פתרון אחד ולא מעניין אותנו אם יש יותר מאחד ולכן ברגע שמצאנו פתרון אנחנו עוצרים את ריצת הפונקציה ומחזירים אותו, לעומת זאת ישנם מקרים שבהם כל הפתרונות האפשריים של הבעיה מעניינים אותנו (אולי כי אנחנו רוצים לבחור את הפתרון הטוב ביותר או לספור את הפתרונות) ולכן לא נעצור את הפונקציה עד שנדע בוודאות שלא קיימים פתרונות נוספים מלבד אלו שכבר ראינו.
ניתן לתאר באופן ציורי את פעולת הפונקציה באמצעות התחלה בצומת נתון של עץ[4] ומעבר על כל הצמתים המקושרים לצומת בזו אחר זו (כאשר לכל אחת מהן עושים את אותה הפעולה) עד שמגיעים לצומת שממנו ודאי שכבר אין פתרון בענף הזה או שמגיעים לצומת שממנו א"א להמשיך לשום כיוון מלבד זה שבאנו בו[5] שהוא פתרון; אם אף אחד מן הקצוות אינו מהווה פתרון אז לבעיה אין פתרון ואם אחד מהם אכן מהווה פתרון אז הדבר תלוי בגרסה של הגישוש הנסוג: אם אנחנו מחפשים פתרון יחיד נפסיק את החיפוש ונחזיר אותו ואם אנחנו מעוניינים בכל הפתרונות האפשריים נמשיך לחפש פתרונות (כלומר נשמור את הנתונים שאנחנו צריכים מהפתרון שמצאנו ואז נחזור חזרה אחורה כאילו לא מצאנו פתרון ונמשיך לחפש במקומות אחרים).
תמונה הייתה שווה כאן אלף מילים.

[3] ערך בוויקיפדיה: פיבונאצ'י.
[4] ייתכן שישנם כמה צמתים שבהם ניתן להתחיל ואז ייתכן שנצטרך לעבור על כולם.
[5] בתורת הגרפים נקרא "עלה".

גישוש נסוג - גרסה ראשונה:

אלגוריתם 4 גישוש נסוג - גרסה 1

הקלט יכול להיות כל דבר.

- אם הקלט הוא פתרון ודאי:
  - החזר את הפתרון.
- אחרת, אם לכל אפשרות להמשיך מן הקלט ודאי שאין פתרון (כי כל אפשרויות ההתקדמות נבדקו והן לא מובילות לפתרון או שאין אפשרויות התקדמות והקלט עצמו אינו פתרון):
  - אם הקלט הוא הקלט המקורי (הראשון שהתקבל ברקורסיה):
    - החזר False.
  - אחרת:
    - חזור אחורה לקלט הקודם והפעל עליו את הפונקציה המקורית (קריאה רקורסיבית).
- אחרת, אם ישנן אפשרויות התקדמות שלא נבדקו:
  - בחר אחת מהן והפעל עליה את הפונקציה המקורית (קריאה רקורסיבית).

גישוש נסוג - גרסה שנייה:

אלגוריתם 5 גישוש נסוג - גרסה 2

הקלט יכול להיות כל דבר.
צור אובייקט ריק x שישמור את הנתונים הנדרשים מכל הפתרונות שנמצא בהמשך.

- אם כל אפשרות להמשיך מן הקלט כבר נבדקה:
  - אם הקלט הוא הקלט המקורי (הראשון שהתקבל ברקורסיה):
    - החזר את x.
  - אחרת, אם הקלט הוא פתרון:
    - עדכן את x ע"פ קלט זה, חזור אחורה לקלט הקודם והפעל עליו את הפונקציה המקורית (קריאה רקורסיבית).
  - אחרת (אם הקלט אינו פתרון):
    - חזור אחורה לקלט הקודם והפעל עליו את הפונקציה המקורית (קריאה רקורסיבית).
- אחרת (אם ישנן אפשרויות התקדמות שלא נבדקו):
  - בחר אחת מהן והפעל עליה את הפונקציה המקורית (קריאה רקורסיבית).

♣ כמובן שניתן לחלק למקרים בסדר אחר[6] אך יש לעבור על כולם.

♣ החלק הכי מסובך בגישוש נסוג הוא החזרה לאחור, פעמים רבות ההתקדמות שלנו מתבטאת ע"י שינוי הקלט[7] (למשל בפתרון סודוקו) ואז כדי שהחזרה לאחור תתבצע בצורה מוצלחת עלינו "לנקות" אחרינו את השטח ולוודא שלא השארנו עקבות, כלומר עלינו להחזיר את המצב לקדמותו ולבטל כל שינוי שעשינו בקלט עד לנקודה שאליה אנחנו חוזרים, זה לא כל כך פשוט.

[6] למשל להכניס בגרסה הראשונה את האפשרות שהקלט הוא פתרון למקרה שבו כל האפשרויות נבדקו כפי שעשינו בגרסה השנייה או להפך: להוציא בגרסה השנייה את המקרה שהקלט הוא פתרון מתוך המקרה שכל האפשרויות נבדקו ולשים אותו בנפרד כפי שעשינו בגרסה הראשונה.

[7] או העתק שלו אם אסור לנו לשנות את הקלט.