

בית הספר להנדסה ומדעי המחשב ע"ש רחל וסלים בנין

מבוא למדעי המחשב 67101

תרגיל 8 - backtracking

להגשה בתאריך 12/12/2018 בשעה 22:00

בתרגיל זה נפתור שתי בעיות בעזרת backtracking. כמו כן נתרגל מעבר על קוד שלא אנחנו כתבנו ומתן ביקורת עליו.

חלק ראשון: סודוקו

אנחנו נפתור בעיה מוכללת של הסודוקו המוכר והאהוב.

לוח סודוקו הוא לוח משבצות עם n שורות ו- n עמודות כאשר n הוא מספר ריבועי (כלומר ריבוע של מספר שלם). הלוח מחולק ל- \sqrt{n} על \sqrt{n} ריבועים בגודל \sqrt{n} על \sqrt{n} משבצות. לדוגמא, עבור $n=9$, הלוח נראה כך:

שימו לב כי בלוח יש 9 שורות ו-9 עמודות והוא מחולק לתשעה ריבועים של 3 על 3.

לוח סודוקו מגיע עם מספרים בחלק מהמשבצות והמטרה היא למלא אותו באופן הבא: בכל שורה, עמודה וריבוע צריך להופיע כל מספר מאחת עד n מבלי לשנות את המספרים המקוריים. חשוב לציין כי כפועל יוצא, אף מספר לא יכול להופיע יותר מפעם אחת באף שורה עמודה או ריבוע.

לקריאה נוספת:

[Sudoku](#)

[סודוקו](#)

[سودوكو](#)

בית הספר להנדסה ומדעי המחשב ע"ש רחל וסלים בנין

עליכם לכתוב את הפונקציה:

```
solve_sudoku(board)
```

המקבלת מערך דו מימדי המייצג לוח סודוקו שחלקו מלא וממלא את אותו לגמרי. תא עם המספר 0 מייצג משבצת ריקה ובכל תא אחר יהיה מספר מ-1 עד n.

כל רשימה בתוך המערך הדו מימדי מייצגת שורה בלוח וכל תא ברשימה מייצג משבצת באותה השורה.

ניתן להניח כי הלוח מגודל n על n עבור n **ריבועי** כלשהו והתאים שכבר מלאים לא מפרים אף אחד מהכללים שפורטו לעיל.

אם יש פתרון ללוח - בסוף ריצת הפונקציה המערך שהתקבל יהיה לוח פתור (ללא אפסים), אחרת, המערך יחזור לצורתו המקורית (כפי שהתקבל בפונקציה). שימו לב כי יכול להיות יותר מפתרון אחד לכל לוח. בנוסף, הפונקציה תחזיר True אם קיים פתרון ללוח ו-False אחרת.

בתרגיל זה אתם אחראים לחלוקת הקוד לפונקציות ולחשיבה על דרך הפתרון.

הערה על בדיקות: פתרון בעיית סודוקו גדולה עלול לקחת זמן רב. על כן כשתכתבו את בדיקותיכם לפונקציה מומלץ לעבוד עם לוחות קטנים, או לוחות שמתחילים כמעט מלאים.

חלק שני: וריאציות על פתרון מוכר

בחלק זה נסתכל על בעיה שפתרנו בתרגול ונפתור אותה בכמה דרכים. בתרגול השבוע ראינו פתרון לבעיית k_subset .

בבעייה זו נקבל שני מספרים n ו-k ונרצה לעבור על כל תתי הקבוצות בגודל k של הקבוצה $\{0, \dots, n-1\}$.

אנו נפתור שלוש וריאציות של הבעיה הזו. שימו לב שאחת הוריאציות נפתרה כבר בתרגול. שימו לב כי בחלק זה מותר לשכפל קוד.

בכל אחד מהפתרונות כל תת קבוצה צריכה להופיע ממיינית, אך הסדר בין תתי הקבוצות לא משנה.

ממשו את שלוש הפונקציות הבאות:

1.

```
print_k_subsets(n,k)
```

המדפיסה את כל תת הקבוצות בגודל k של הקבוצה $\{0, \dots, n-1\}$. כשברקורסיה נגיע לתת קבוצה עם k איברים יש לשמור אותה ברשימה ממיינית מהאיבר הקטן לגדול, ולהדפיס את הרשימה. יש להדפיס כל תת קבוצה בשורה נפרדת. למשל, קריאה ל:

```
print_k_subsets(3,2)
```

תדפיס את השורות הבאות:

```
[0, 1]
```

```
[0, 2]
```

בית הספר להנדסה ומדעי המחשב ע"ש רחל וסלים בנין

[1, 2]

שימו לב כי ייצור רשימה עם כל תתי הקבוצות והדפסתה לאחר מכן משתמש בהרבה מקום שלא לצורך.

.2

`fill_k_subsets(n,k,lst)`

אשר תקבל זוג מספרים n, k ורשימה ריקה `lst`, ובסוף הריצה תמלא ב-`lst` את כל תתי הקבוצות בגודל k של הקבוצה $\{0, \dots, n-1\}$, כלומר את כל תתי הקבוצות שהודפסו בסעיף הקודם. בסעיף זה בכל פעם שנמצא תת קבוצה בגודל k נשמור אותה בתוך רשימה ואז נכניס את הרשימה ל-`lst`. בסעיף זה אין להחזיר ערכים לאורך כל הרקורסיה.

.3

`return_k_subsets(n,k)`

המחזירה רשימה של רשימות. בכל רשימה יהיו בדיוק k איברים שונים מתוך הקבוצה $\{0, \dots, n-1\}$, כמו בסעיפים הקודמים..

בפתרון סעיף זה אין להעביר רשימות כארגומנטים כלל, או איברים דומים כגון `tuple`, `set` וכו'.

חלק שלישי: Code Review

השבוע כל אחד מכם יקבל שני פתרונות של חבריכם לתרגיל 6 ותתבקשו לעשות להם `code review` (סקר קוד).

סקר קוד הוא הליך מקובל מאד בחברות תוכנה בו מתכנת אחד קורא קוד של מתכנת אחר ומחווה את דעתו עליו. לסקר קוד מטרות רבות, ביניהן מציאת שגיאות, שיפור הקוד, תכנון קוד נכון יותר, למידה מניסיונם של מתכנתים אחרים ועוד.

במודל השבוע יהיה לינק ל-Code Review ובו תקבלו גישה לשני פתרונות לתרגיל 6. עליכם לקרוא את הפתרון, להבין אותו ולהעביר עליו ביקורת. בביקורת יש להתייחס להיבטים הבאים:

- תכנון הקוד וחלוקתו לפונקציות.
- קריאות הקוד - האם היו חלקים שהיו קשים להבנה?
- האם הקוד מודולרי וקל לשינוי? למשל, אילו היינו רוצים להוסיף אופציות לחלק מהתפריטים כמה שינויים היו נגזרים בקוד.
- שגיאות מפורשות בפתרון התרגיל.

חשוב לציין שלא מספיק לכתוב מה נעשה לא נכון, אלא צריך גם לכתוב איך היה ניתן לפתור את הבעיה. לכל הערה יש לכתוב את מספר השורה ושם הפונקציה הרלוונטיים. כמו כן, כתבו על דברים שנעשו טוב בפתרון התרגיל ועל דברים שאתם למדתם מקריאתו. זאת אומרת שבכל מקרה יש לעבור על כל תרגיל ולהעיר הערות מפורטות על אילו חלקים היה צריך לשנות ואיך, ועל אילו חלקים נעשו היטב לדעתכם ולמה.

בית הספר להנדסה ומדעי המחשב ע"ש רחל וסלים בנין

שימו לב שהבודקים יבדקו את סקירות הקוד שכתבתם וייתנו עליהן ציון. עם זאת, תוצאת הסקירה לא תשפיע על ציון התרגיל שנסקר. בסיום הגשת תרגיל זה, כל אחד יקבל את הסקירות שנעשו על הקוד שלו. אנו ממליצים בחום לעבור על ההערות ולנסות ללמוד מחבריכם לקורס.

נהלי הגשה

הלינק להגשה של התרגיל הוא תחת השם: 8ex

בתרגיל זה עליכם להגיש את הקבצים הבאים:

1. ex8.py – עם המימושים שלכם לפונקציות .
2. README (על פי פורמט ה-README לדוגמא שיש באתר הקורס, ועל פי ההנחיות לכתיבת README המפורטות בקובץ נהלי הקורס).

יש להגיש קובץ zip הנקרא ex8.zip המכיל בדיוק את שני הקבצים הנ"ל.

כמו כן, עליכם לכתוב את ה-code review שלכם בלינק המתאים ב-moodle

בהצלחה