**Read me – מבוא לבינה מלאכותית, ממן 15**

דולב חגבי 316016757

הערה: קיבלתי אישור להגיש מאוחר. האישור ניתן מכוון שגיליתי שיש דף שני בממן (עם הסטנדרטים) יומיים לאחר ההגשה...

**פתרון בעיות קריפטואריתמטיות כבעיות סיפוק אילוצים.**

**הסבר על האלגוריתם:**

לפני ההרצה:

ראשית, נסדר את הבעיה שקיבלנו על ידי פונקציית **organizeProblem** שמכניסה את המילים לרשימה דו מימדית ומגדירה את המקומות הריקים כאפסים. לדוגמה, עבור SEND MORE MONEY:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| D | N | E | S | 0 |
| E | R | O | M | 0 |
| Y | E | N | O | M |

**makeCSP: יוצרת רשימת אילוצים**

הפונקציה מקבלת את הבעיה המסודרת ויוצרת את רשימת האילוצים הבאה:

* Y = (D+E) % 10

נרצה שהאות האחרונה של המילה השלישית תהיה היחידות של תוצאת החיבור בין האותיות של שתי המילים.

* E = (N + R + (D+E) / 10) % 10

נרצה שהאות האחת לפני האחרונה של המילה השלישית תהיה היחידות של תוצאת החיבור בין האותיות של שתי המילים + העשרות של החיבור הקודם.

נמשיך הלאה עד לספרה האחרונה שהיא העשרות של תוצאת החיבור הקודם, והיא תוצאה של החיבור הקודם וכן הלאה, כך שהספרה האחרונה מורכבת מכל האותיות שהיו עד כה.

בנוסף לזה נרצה :

* החיבור של 0+0 באות הראשונה יהיה שווה ל1 (העשרות של החיבור של האותיות הקודמות)
* לא נסכים לכך שהאות הראשונה במילה תהיה אפס (M וS).

פונקציות עזר:

**MRV: נבחר אות שאותה נשבץ**

ראשית נחפש אילוצים אונארים. לדוגמה: M==1 אם קיים כזה נבחר אותו.

אם לא קיים נחפש את האיבר בעל הכי הרבה אילוצים (מהגדרת MRV). ננמש זאת על ידי מעבר על רשימת האילוצים ויצירת מילון של - {אות: מספר הפעמים שמופיעה באילוצים}. נבחר את האות שמופיעה הכי הרבה פעמים.

**LCV: נבחר שיבוץ לאות שנבחרה**

מהגדרת LCV נרצה לבחור מספר שיצמצם את מספר האילוצים. נשים לב שככל המספרים גדולים יותר, כך סביר שתעבור יחידה קדימה לספרה הבאה ותגדיל את האילוצים (לדוגמה, אם בכל פעם תועבר יחידה קדימה אז המספר האחרון יהיה תלוי בכל המספרים שהיו עד כה, לעומת שני המספרים שמעליו בלבד).

לכן נסדר את רשימת האופציות של האות שנבחרה לפי גודל, כדי שהמספר הנמוך ביותר יבחר.

**AC3: צמצום הטווח של רשימת האופציות**

הפונקציה עוברת על הבעיה הנוכחית ויוצרת רשימת אילוצים חדשה באמצעות makeCSP. היא עוברת על האילוצים ("הקשתות"), שולפת מתוכם את האותיות ("הקודקודים") ומכניסה אותם לתור.

עוברים על האותיות שבתור ובודקים כל אופציה ברשימת האופציות של אותה אות. אם האופציה יוצרת סתירה, מסירים את האופציה מהרשימה.

AC3 מחזירה *FIALURE* אם נוצרת רשימה ריקה, ורשימת אופציות מעודכנת אחרת.

פונקציה מרכזית:

**backtrack : החלק העיקרי באלגוריתם**

מקבלת את הבעיה הנוכחית לאחר ההשמות (רשימה שבהתחלה מכילה את הבעיה המסודרת ולאחר כל השמה מוחלפות בו האותיות המושמות למספרים), ורשימת אופציות (מילון- {אות: רשימה של אופציות רלוונטיות להשמה})

אם הבעיה הושלמה, כלומר הרשימה מכילה מספרים בלבד – הפתרון יודפס (יחד עם מספרו).

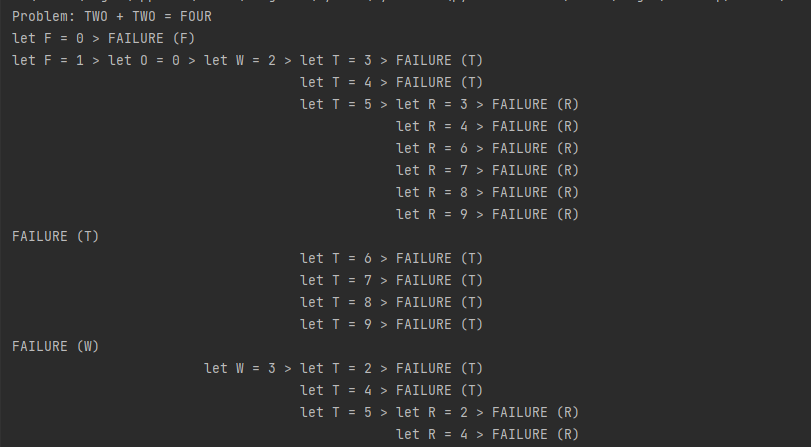
נוודא שרשימת האופציות תקינה על ידי AC3, לאחר מכן נבחר אות על ידי MRV, נסדר את רשימת האופציות לאותה אות על ידי LCV, ונעבור בלולאה על רשימת האפשרויות המעודכנת של אותה אות (כמובן שאם ימצא פתרון תוך כדי ריצה נמשיך איתו):

נעדכן את הבעיה על ידי הפונקציה **updateAns** (שמחליפה את האותיות במספר שנבחר ומוודאת שהחלפה תקינה) ונבדוק גם את תקינות האילוצים (כלומר, שאין רשימה ריקה) על ידי **AC3**.

אם הייתה שגיאה הפונקציה תחזיר *FIALURE* ותמשיך למספר הבא ברשימה באיטרציה הקודמת.

אם הכל תקין תעבור לאות הבאה להשמה על ידי קריאה רקורסיבית של הפונקציה עם הבעיה המעודכנת ורשימת האופציות המעודכנות.

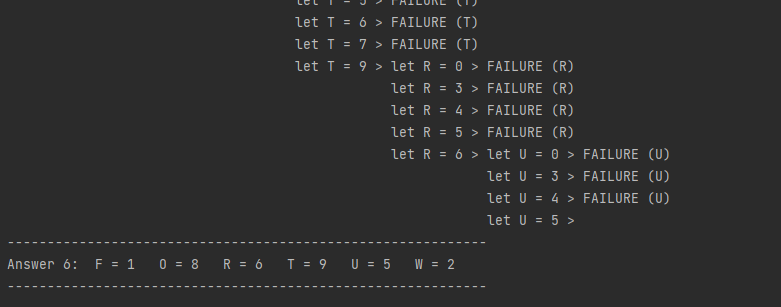
**דוגמה #1: TWO TWO FOUR**



ניתן לראות את ההשמות, ואת בחירת היוריסטיקות בכל איטרציה, כמו כן את החזרות לאחור ואת ההשמה מחדש.

...

דוגמה לאחד הפתרונות שנמצאו:



...

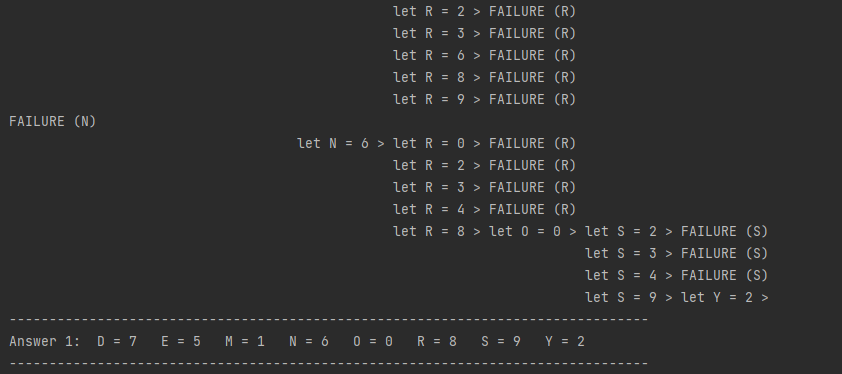


**דוגמה #2: SEND MORE MONEY**

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

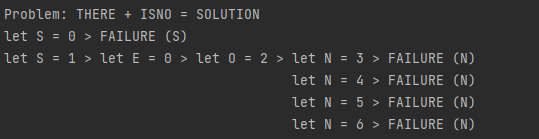
...



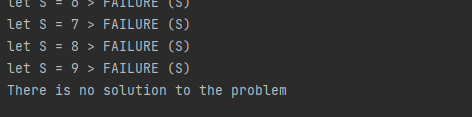


נמצא פתרון יחיד מכוון שהוגדר ש M!=0

**דוגמה #3: THERE ISNO SOLUTION**

****

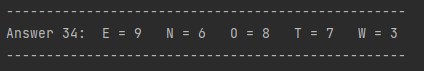
**...**

****

**דוגמה #4: ONE ONE TWO**

**תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי**

****

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי