

**VIETNAM NATIONAL UNIVERSITY HO CHI MINH CITY**  
**VNUHCM - UNIVERSITY OF SCIENCE**  
**FACULTY: INFORMATION TECHNOLOGY**

---o0o---



# **REPORT**

## **PROJECT 01: CRYPTARITHMETIC PROBLEM IN AI**

**COURSE: CSC14003 – INTRODUCTION TO ARTIFICIAL INTELLIGENCE**

**TEACHER: NGUYỄN NGỌC THẢO**

**MEMBERS: LÊ VŨ HUY – 19127421**

**NGUYỄN QUANG PHÚ – 19127507**

**LÂM HOÀNG PHÚC – 19127512**

**CLASS: 19CLC1 – TERM III/2020-2021**

**07-2021**

**1. Information about us:**

No.	Student ID	Full name
1	19127421	Lê Vũ Huy
2	19127507	Nguyễn Quang Phú
3	19127512	Lâm Hoàng Phúc

**2. Assignment plan:**

No.	Task	Implement
1	Brainstorm ideas, design algorithms to solve problems	Whole team
2	Design algorithm for each level.	Whole team
3	Backtracking algorithm (All level)	Lê Vũ Huy
4	Genetic algorithm (All level)	Lâm Hoàng Phúc
5	Test	Nguyễn Quang Phú
6	Report	Whole team

**3. Environment:**



Visual Studio Code



Python 3.9

#### 4. Self-assessment:

No.	Requirement	Note	Completemet
1	Level 1: Solve the addition equation or subtraction equation with 2 operands.	Use two algorithms (Backtracking algorithm and Genetic algorithm) running concurrently (threading) to solve the problem. If either one of the two is found, the program is stopped and otherwise the message is no result.	100%
2	Level 2: Solve any kind of equation with multiple operands and single operator (+ -).		100%
3	Level 3: Solve any kind of equation with multiple operands and operators (+ - ). We use the parentheses ( ) are used to specify the order of operations.		100%
4	Level 4: Solve level 3 with the multiplication equation (*).		100%
5	Generate at least 5 test cases for each level with different attributes such as size, time to solve, output type. Describe them in the experiment section of your report.		100%
6	Report your algorithms, experiments with some reflection or comments.		100%
<b>Total</b>			<b>100%</b>

#### 5. Algorithms

##### 5.1 File input

We read only the first line of input.txt. Then we convert this line from string into a list operands (example: ["SEND", "+", "MORE", "=", "MONEY"]).

##### 5.2 Data structure

["D", "E", "Y", "N", "R", "O", "S", "M", ""] (example).

Index of the character in the list is the value (D = 0).

##### 5.3 Genetic algorithm

We use genetic algorithm to solve the problem in another thread. Executing this algorithm can randomly solve the problem faster than backtracking.

1. Extract distinct letters from input strings and put them in a list named L.
2. Repeat the following until the desired population size is reached:
  - a. For each letter in L do the following:
    - i. Generate a random number between 0 and 9.

- ii. If the random number is equal to zero and the letter is the beginning letter of the words (which should not be zero) go to iv.
  - iii. If the cell that is corresponding the random number is empty, put the letter into that cell, else go to step i.
  - iv. If the current letter is the last letter of L and the length of L is 10 then generate a random number between 1 and 9 and exchange the place of the letter from that cell to the index of zero and the current letter to that position. Keep doing it until the second letter is not a beginning letter of the words. Else go to step i.
- b. Calculate the fitness of this new individual.
  - c. Add the individual into the right place in the inner population list.
3. Then we sort the current generations with fitness (fitness == abs(left side of equal sign - right side of equal sign)).
  4. If generation at index 0 has fitness equal to zero. Then we return the solution.
  5. We generate next generations:
    - a. Get the chosen parent and generate two random numbers.
    - b. If the two random numbers are not the same or not indicating two empty cells in the parent and will not lead us to numbers beginning with zeros, exchanges the content of the cells corresponded by the random numbers. Else go to step a.
    - c. Calculate the fitness of the new individual, put it in the right place in the inner population and remove the worst individual from the population.
  6. Go to step 3.

#### **5.4 Backtracking algorithm**

- a. We use a backtracking algorithm in the main thread to solve this problem. For each assignment, we check the current assignment with the most left character (in case the number has more than one digit ) and columns. We ensure that every assignment has to be completed column by column.
- b. Example ["SEND", "+", "MORE", "=", "MONEY"]

We assign 0 to D. because D is at index 0 in data structure so we assign 1 to E. Then we assign 2 to Y, but now, we fully assign the most right column so we check whether this column is completed with constraint  $(D+E+carry)\%10=Y$  (or  $D+E \% 10 = Y$  with asterisk). We keep repeating until solution is found (we also check whether S and M are 0 when we assign value to it).

## 6. Experiments

Level	Test case	Size	Output	Running time (avg) (ms)
1	SEND+MORE=MONEY	8 characters, 1 operator	DEMNORSY 75160892	247.4
	DAN+NAN=NORA	5 characters, 1 operator	ADNOR 29104	22.0
	BASIC+LOGIC=PASCAL	9 characters, 1 operator	ABCGIOPS 062354718	40.7
	BASE+BALL=GAMES	7 characters, 1 operator	ABEGLMS 4731598	26.4
	TWO+TWO=FOUR	6 characters, 1 operator	FORTUW 148763	20.2
<p><b>Comment:</b> At level 1, the test cases run pretty fast (&lt; 0.5s). In case 1, <b>Genetic</b> runs faster than <b>Backtracking</b> and vice versa in the remaining cases.</p>				
2	SO+MANY+MORE+MEN+SE EN+TO+SAY+THAT+THEY+ MAY+SOON+TRY+TO+STAY +AT+HOME+SO+AS+TO+SE E+OR+HEAR+THE+SAME+O NE+MAN+TRY+TO+MEET+T HE+TEAM+ON+THE+MOON +AS+HE+HAS+AT+THE+OT HER+TEN=TESTS	10 characters, 1 operator	AEHMNORS TY 9642015387	24220.4
	STAY+AT+HOME=TESTS	8 characters, 1 operator	AEHMOSTY 24973510	25.4
	THREE-MEN-MEET=TESTS	7 characters, 1 operator	EHMNRST 3417506	148.8
	THE+SAME+ONE+MAN+TR Y+TO+MEET+THE+TEAM+O N+THE+MOON=TESTS	10 characters, 1 operator	AEHMNORS TY 9064158723	264.6

*Course: CSC14003 – Introduction to Artificial Intelligence*  
*Class 19CLC1 – Term III/2020-2021*

	HE+HAS+AT+THE+OTHER+TEN=TESTS	8 characters, 1 operator	AEHNORST 60297318	78.2
<p><b>Comment:</b> At level 2, the test cases all run pretty fast (&lt; 0.5s) except for case 1 (24s), depends on the number of operators.</p>				
3	ON+THE+MOON-AS+HE+HAS+AT+(THE+OTHER-TEN)=TESTS	9 characters, 3 operators	AEHMNORS T 819205436	75.15
	AI+(NEVER-DIES)=TESTS	9 characters, 3 operators	ADEINRSTV 391856240	1340.5
	SEND+(MORE+MONEY)-OR+DIE=NUOYI	10 characters, 3 operators	DEIMNORS UY 1295630874	214.2
	THEY+MAY+SOON+(TRY+T O-STAY+AT+HOME)=TESTS	10 characters, 3 operators	AEHMNORS TY 6297830415	314.5
	THREE-(MEN+MEET)=TESTS	7 characters, 3 operators	EHMNRST 1543608	70.8
<p><b>Comment:</b> At level 3, the test cases all run pretty fast (&lt; 1s). In case 4, <b>Genetic</b> runs faster than <b>Backtracking</b> and vice versa in the remaining cases, depends on the number of operators.</p>				
4	AB*C=DE	5 characters, 1 operator	ABCDE 12560	21.5
	AB*CD=DEF	6 characters, 1 operator	ABCDEF 123408	19.7
	MAD*MAN=ASYLUM	8 characters, 1 operator	ADLMNSUY 47268159	136.9
	ABCD*EFGHI=GJJGGFBF	10 characters, 1 operator	ABCDEFGHI J 1234567890	375.3

	ABCE*AEBF=AHHDFAT	8 characters, 1 operator	ABCDEFHT 12345680	253.9
<b>Comment:</b> At level 3, the test cases all run pretty fast (< 0.5s).				

## 7. References

<https://www.codeproject.com/Articles/176768/Cryptarithmic>

<https://www.youtube.com/watch?v=aZblDSAx4cg> (Cryptarithmic problem for CSPs)

<https://gist.github.com/vo/2481737> (backtracking algorithm)

<https://staff.fnwi.uva.nl/m.mazloom/Papers/mazloom-milad.pdf?fbclid=IwAR3yTepbgIOFfjl2HJqpRIaPdQDR5EWZWQYsscH17ZU497RZo2SddiwBjo> (Genetic algorithm)