

컴퓨팅적 사고와 문제해결

과제 2

중앙대학교 소프트웨어학부 19학번 유용민
(391040342)

문제 1. 자기소개서 이름 추출

```
1 # 사용자로부터 문장을 입력받고 이를 공백으로 구분하여 리스트 형식으로 저장합니다.
2 inputStr = input("문장을 입력하세요 : ")
3 inputStrList = inputStr.split()
4
5 for i in inputStrList:
6     # 주어진 자기소개 문장 형식은 "저는 홍길동입니다.", "저는 홍길동이라고 합니다"와 같은 형식이므로
7     # split()으로 분리된 구절 중 "입니다" 또는 "이라고" 가 포함된 구절을 찾아
8     # 해당 구절에서 "입니다" 또는 "이라고" 의 앞부분까지를 이름으로 간주하고 이를 slice하여 출력함
9     if i.count("입니다"):
10         print(i[:i.find("입니다")])
11         break
12     elif i.count("이라고"):
13         print(i[:i.find("이라고")])
14         break
```



```
C:\Windows\system32\cmd.exe
문장을 입력하세요 : 저는 홍길동이라고 합니다.
홍길동
문장을 입력하세요 : 저는 심청입니다.
심청
문장을 입력하세요 :
```

[https://github.com/yymin1022/CAU-Computational-Thinking-and-Problem-Solving-Assignment2/blob/master/문제 1.py](https://github.com/yymin1022/CAU-Computational-Thinking-and-Problem-Solving-Assignment2/blob/master/문제%201.py)

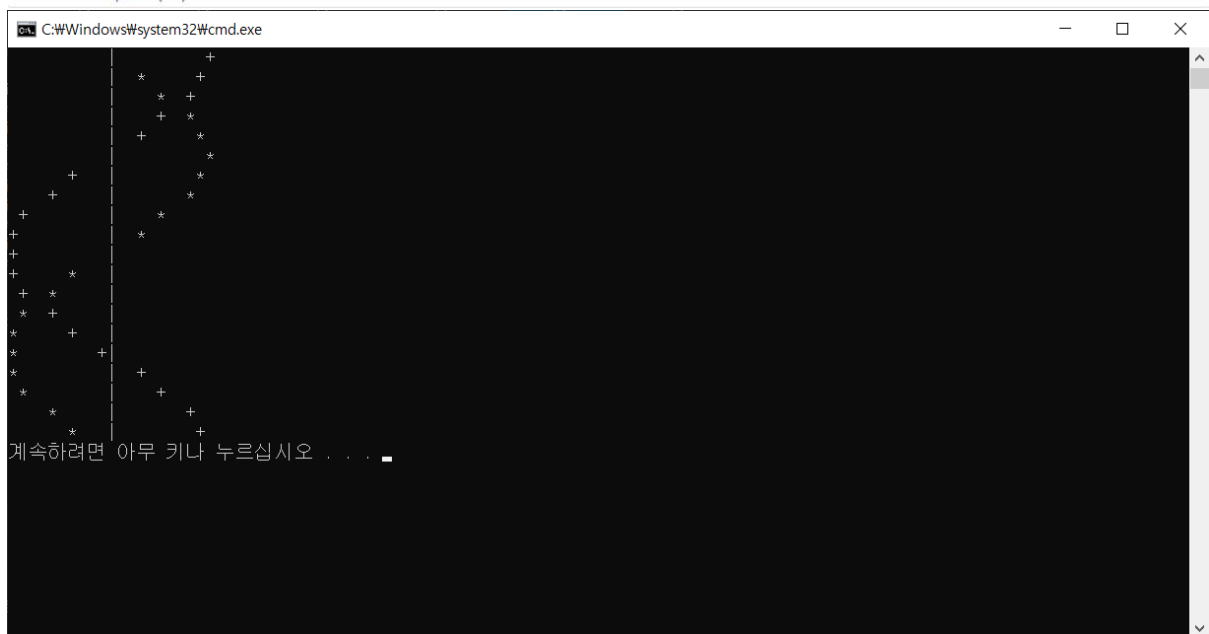
문제에서 입력되는 문장은 "저는 홍길동이라고 합니다..", "저는 심청입니다"와 같은 형태로 제한되어 있습니다.

이에 따라 먼저 입력된 문장을 공백 단위로 쪼개 뒤 각 부분에 "입니다" 또는 "이라고"가 포함되는지의 여부를 확인합니다.

"입니다" 또는 "이라고"가 포함된 경우에는 해당 부분에서 "입니다"와 "이라고"의 앞 까지를 slice 하여 이를 이름으로 출력합니다.

문제 2. 사인 코사인 그래프 그리기

```
1 from math import sin, cos, pi
2
3 for i in range(20):
4     # 21개의 index를 갖는 리스트 선언
5     linePrint = [" ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " "]
6     # sin 값과 cos 값을 계산해 각 값의 범위가 0 ~ 20 사이가 되도록 식 조절 후
7     # 계산된 값을 int 정수형으로 변환해 해당 index의 리스트에 저장
8     linePrint[(int(10 + 10 * sin(i * pi / 10)))] = "*"
9     linePrint[(int(10 + 10 * cos(i * pi / 10)))] = "+"
10    # 21개의 index 중 중앙값인 index 10 값에 기준선을 출력하도록 지정
11    linePrint[10] = "|"
12    # 리스트의 각 요소를 출력하도록 for 반복문 이용
13    for j in linePrint:
14        print(j, end=" ")
15    print("")
```



<https://github.com/yymin1022/CAU-Computational-Thinking-and-Problem-Solving-Assignment2/blob/master/문제 2.py>

21 개의 칸으로 이루어진 리스트에 0 부터 360 도 까지의 사인 값, 코사인 값을 일정한 간격으로 20 개 구해 리스트의 요소로 저장합니다.

리스트의 11 번째 요소로는 |를 저장해주어 그래프의 기준선이 되도록 합니다.

문제 3. 로또 번호 생성기

```
1 import random
2
3 lottoList = [] # 1000회의 로또 진행 후 각 결과가 저장될 리스트
4 lottoCount = [] # 1부터 45까지의 값이 각각 나온 횟수가 저장될 리스트
5 lottoCountSort = [] # lottoCount 리스트가 Sort되어 저장될 리스트
6 lottoMax = [0, 0, 0, 0, 0, 0] # 가장 빈도가 높은 6개의 숫자가 저장될 리스트
7
8 def lotto_generator():
9     lottoNum = []
10    # 번호 추첨을 6번 반복
11    for g in range(6):
12        # 1부터 45 사이의 범위에서 정수 난수를 생성
13        randNum = random.randrange(1, 46)
14        # 이미 뽑힌 난수인지 확인하고, 중복된 경우 다시 난수 생성
15        while randNum in lottoNum:
16            randNum = random.randrange(1, 46)
17        # 생성된 난수를 lottoNum 리스트에 append
18        lottoNum.append(randNum)
19        # 난수가 출현된 횟수를 1 증가시킴 : 생성된 난수에서 1을 뺀 값이 리스트의 index 값
20        lottoCount[lottoNum[g]-1] = lottoCount[lottoNum[g]-1] + 1
21    # 생성된 6개 번호가 저장된 리스트인 lottoNum을 lottoList 리스트에 element로 추가
22    lottoList.append(lottoNum)
23
24    # 각 숫자가 출현된 빈도를 기록하기 위한 리스트인 lottoCount 리스트에 0을 45개 추가하여
25    # 이후에 숫자가 출현될 때 마다 1씩 더하도록 해줌
26    for i in range(45):
27        lottoCount.append(0)
28
29    # 번호 1세트(6개)를 추첨하는 함수인 lotto_generator() 함수를 1000회 실행
30    for j in range(1000):
31        lotto_generator()
32
33    # lottoCount 리스트에 저장된 각 숫자의 빈도를 출력
34    # index값에 1을 더하면 해당 숫자가 구해짐 : index 0에 저장된 값은 1이 출현한 횟수
35    for count in range(45):
36        print("%d가 나온 횟수는 %d회 입니다."%(count+1, lottoCount[count]))
37
38    # lottoCount 리스트에서 가장 큰 값 6개를 추출하는 for문
39    # lottoCount에 저장된 값은 빈도 수 이므로
40    # 가장 큰 값의 index에 1을 더한 값을 lottoMax 리스트에 저장
41    for k in range(6):
42        for l in range(45):
43            if k == 0:
44                if lottoCount[l] > lottoCount[lottoMax[k] - 1]:
45                    lottoMax[k] = l + 1
46            else:
47                if lottoCount[l] > lottoCount[lottoMax[k] - 1] and lottoCount[l] <= lottoCount[lottoMax[k] - 1]:
48                    if (l + 1) not in lottoMax:
49                        lottoMax[k] = l + 1
50
51    # lottoMax 리스트를 이 주의 추천 번호로 출력
52    print("이 주의 추천 로또 번호는", lottoMax, "입니다.")
```

```
C:\Windows\system32\cmd.exe
18가 나 123회
19가 나 132회
20가 나 132회
21가 나 143회
22가 나 140회
23가 나 121회
24가 나 129회
25가 나 126회
26가 나 135회
27가 나 127회
28가 나 154회
29가 나 98회
30가 나 132회
31가 나 140회
32가 나 130회
33가 나 122회
34가 나 135회
35가 나 134회
36가 나 154회
37가 나 149회
38가 나 129회
39가 나 111회
40가 나 152회
41가 나 143회
42가 나 143회
43가 나 140회
44가 나 125회
45가 나 120회
이 주의 추천 로또 번호는 [7, 28, 36, 40, 1, 5] 입니다.
계속하려면 아무 키나 누르시오 . . .
```

<https://github.com/yymin1022/CAU-Computational-Thinking-and-Problem-Solving-Assignment2/blob/master/문제 3.py>

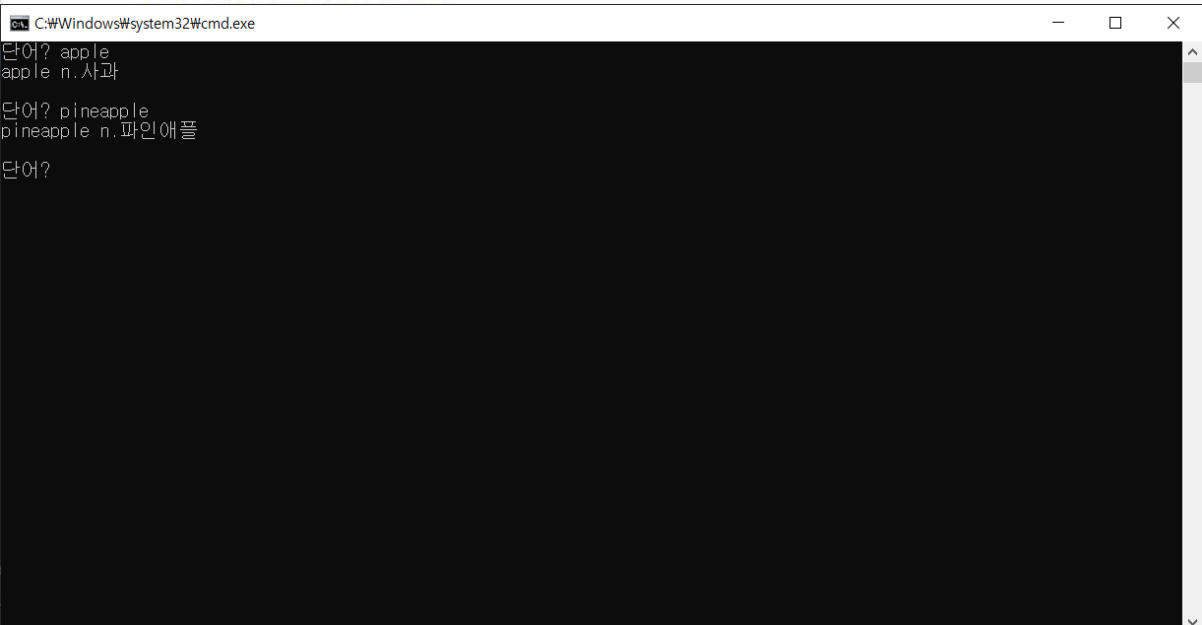
lotto_generator() 함수는 실행될 때 마다 1~45 사이의 범위에서 6 개의 난수를 중복되지 않도록 생성해 이를 리스트로 저장합니다.

이 함수를 1000 번 실행하면 lottoList 리스트에 1000 개의 로또번호 세트가 저장됩니다.

이후에는 각 숫자가 출현한 횟수를 구해 lottoCount 리스트에 저장하고, 이 리스트에서 가장 많이 출현한 6 개의 수를 구해 이 주의 추천 번호로 출력해줍니다.

문제 4. 전자사전 만들기

```
1 while 1:
2     #단어가 사전 내에 존재하는지 여부를 체크하기 위한 boolean 변수
3     existWord = False
4     inputWord = input("단어? ")
5     dictFile = open("dict_test.TXT", "r")
6     while 1:
7         # 입력된 단어가 사전 내에 존재하는 지 확인하는 반복문
8         # 사전을 모두 탐색한 뒤에는 break하여 반복문 종료
9         line = dictFile.readline()
10        if not line: break
11        # 사전이 "영어 : 품사&뜻" 형식으로 저장되어 있으므로
12        # :을 기준으로 사전의 각 줄을 split하면
13        # index값이 0인 부분에 영단어+공백, 1인 부분에 품사&뜻 이 저장됨
14        #rstrip()과 lstrip()함수로 각각의 오른쪽 또는 왼쪽의 공백을 제거하고
15        # 입력된 단어와 읽어낸 사전 줄의 영단어가 일치하는 경우
16        # 해당하는 단어의 뜻 부분을 출력함
17        lineWord = line.split(":")[0].rstrip()
18        if lineWord == inputWord:
19            print(line.split(":")[1].lstrip())
20            existWord = True
21    dictFile.close()
22    if existWord != True:
23        print("사전에 존재하지 않는 단어입니다.")
```



```
C:\Windows\system32\cmd.exe
단어? apple
apple n. 사과

단어? pineapple
pineapple n. 파인애플

단어?
```

[https://github.com/yymin1022/CAU-Computational-Thinking-and-Problem-Solving-Assignment2/blob/master/문제 4py](https://github.com/yymin1022/CAU-Computational-Thinking-and-Problem-Solving-Assignment2/blob/master/문제%204.py)

먼저 단어를 입력 받은 뒤 해당 단어가 사전에 존재하는지 확인합니다. 확인된 이후에는 해당 단어의 뜻 부분을 출력해줍니다.

문제 5. 끝말 잇기

```
1 lastWordList = []
2 lastWord = ""
3
4 # 첫번째 시작 단어를 입력받음
5 while 1:
6     inputWord = input("단어를 입력하세요 ")
7
8     # 단어의 길이가 5가 아닌 경우 경고문 출력
9     if len(inputWord) > 5:
10         print("단어가 길어요.")
11         continue
12     elif len(inputWord) < 5:
13         print("단어가 짧아요.")
14         continue
15
16     dataFile = open("dict_test.TXT", "r")
17     existWord = False
18     isNoun = False
19     # 사전에 존재하는 단어인지, 명사인지의 여부를 확인하는 반복문
20     while True:
21         line = dataFile.readline()
22         if not line: break
23         if line.split()[0] == inputWord:
24             existWord = True
25             if line.split(":")[1].find("n.") != -1:
26                 isNoun = True
27     dataFile.close()
28
29     if existWord == False:
30         print("사전에 존재하지 않는 단어입니다.")
31         continue
32
33     if isNoun == False:
34         print("명사가 아닙니다.")
35         continue
36
37     # 모든 조건에 부합하는 단어인 경우 첫번째 단어로 지정
38     # 입력된 단어를 기록하는 lastWordList에 저장하고
39     # 다음 단어와의 이어지는지 여부 확인을 위해 lastWord String 변수에 저장
40     lastWord = inputWord
41     lastWordList.append(lastWord)
42     break
43
44 while 1:
45     # 첫 단어 입력값과 같은 알고리즘이니,
46     # 이전 단어의 끝 글자와 입력한 단어의 첫 글자가 일치하는 지 여부,
47     # 이전에 입력된 적이 없는 단어인지 여부를
48     # 확인하는 조건문이 추가됨
49     print(lastWord, end=" ")
50     inputWord = input("끝말잇기? ")
51     if len(inputWord) > 5:
52         print("단어가 길어요.")
53         continue
54     elif len(inputWord) < 5:
55         print("단어가 짧아요.")
56         continue
57     elif inputWord[0] != lastWord[-1]:
58         print("입력한 단어의 첫 글자가 이전 단어의 끝 글자와 다릅니다.")
59         continue
60
61     for i in lastWordList:
62         if inputWord == i:
63             print("이미 입력한 단어입니다.")
64             continue
65
66     dataFile = open("dict_test.TXT", "r")
67     existWord = False
68     isNoun = False
69     while True:
70         line = dataFile.readline()
71         if not line: break
72         if line.split()[0] == inputWord:
73             existWord = True
74             if line.split()[2][0] == "n":
75                 isNoun = True
76     dataFile.close()
77
78     if existWord == False:
79         print("사전에 존재하지 않는 단어입니다.")
80         continue
81
82     if isNoun == False:
83         print("명사가 아닙니다.")
84         continue
85
86     lastWordList.append(inputWord)
87     lastWord = inputWord
```

```
C:\Windows\system32\cmd.exe
단어를 입력하세요 apple
apple 끝말잇기? eagle
eagle 끝말잇기? error
error 끝말잇기? orange
단어가 길어요
error 끝말잇기? rose
단어가 짧아요
error 끝말잇기? eagle
이미 입력한 단어입니다.
error 끝말잇기? _
```

[https://github.com/yymin1022/CAU-Computational-Thinking-and-Problem-Solving-Assignment2/blob/master/문제 5.py](https://github.com/yymin1022/CAU-Computational-Thinking-and-Problem-Solving-Assignment2/blob/master/문제%205.py)


입력된 단어를 시작으로 끝말잇기를 진행합니다.

사전에 존재하는 단어인지, 명사인지, 5 글자인지의 여부를 확인한 후 다음 단어를 입력 받도록 합니다.

두번째 단어부터는 이전에 입력된 기록이 없는 단어인지를 추가로 확인하는 조건문이 있습니다.

문제 6. 성적 처리하기

```
1 scoreFile = open("SCORE.csv", "r")
2
3 listStudents = []
4 listSums = []
5
6 # CSV 파일로부터 각 학생의 성적을 읽어들이고
7 # 학번, 국어점수, 영어점수, 수학점수, 총점 의 데이터를
8 #listScore 리스트에 저장
9 while True:
10     listScore = [0, 0, 0, 0, 0, 1]
11     currentLine = scoreFile.readline().replace("\n", "")
12     if not currentLine: break
13     listScore[0] = int(currentLine.split(",")[0])
14     listScore[1] = int(currentLine.split(",")[1])
15     listScore[2] = int(currentLine.split(",")[2])
16     listScore[3] = int(currentLine.split(",")[3])
17     listScore[4] = listScore[1] + listScore[2] + listScore[3]
18     listSums.append(listScore[4])
19
20 # 완성된 학생 한명의 listScore 리스트를 listStudent 리스트의 element로 추가
21 listStudents.append(listScore)
22
23 # 각 학생의 총점이 저장된 listSum 리스트를 내림차순으로 정렬
24 listSums.sort()
25 listSums.reverse()
26
27 # 정렬된 listSum 함수에 따라 각 학생의 석차를 지정
28 # 점수가 낮은 학생일수록 기쁜 석차값인 1로부터 1씩 증가시킴
29 for i in range(len(listSums)):
30     for j in listStudents:
31         if listSums[i] == j[4]:
32             j[5] = i + 1
33
34 print(" 번호   국어   영어   수학   총점   석차")
35 for k in listStudents:
36     for l in range(6):
37         print("%3d"%k[l], end=" ")
38     print("")
```



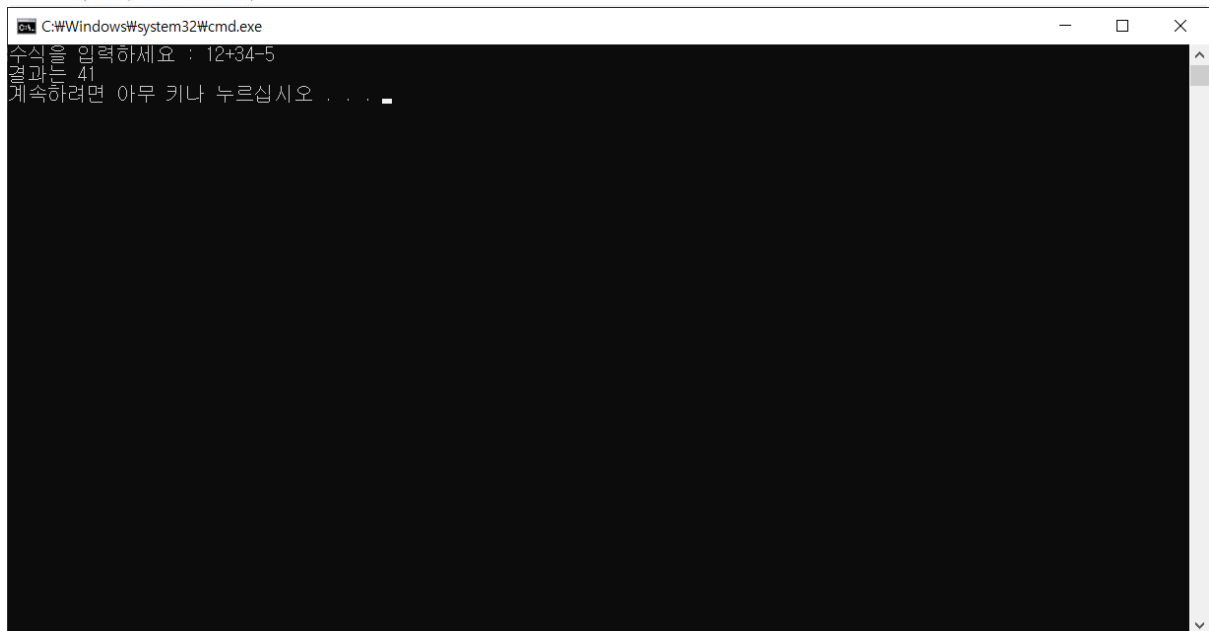
번호	국어	영어	수학	총점	석차
20170101	90	80	70	240	4
20168372	90	95	90	275	2
20180392	100	100	90	290	1
20173882	95	80	75	250	3
계속하려면 아무 키나 누르십시오 . . .					

[https://github.com/yymin1022/CAU-Computational-Thinking-and-Problem-Solving-Assignment2/blob/master/문제 6.py](https://github.com/yymin1022/CAU-Computational-Thinking-and-Problem-Solving-Assignment2/blob/master/문제%206.py)

CSV 파일을 읽어 들이고 이를 ,(Comma)에 따라 잘라낸 뒤 리스트화 하고, 총점에 따라 석차를 지정해주는 코드입니다.

문제 7. 수식 계산기

```
1 # 수식을 입력받고 이를 +를 기준으로 split하여 splitedPlus 리스트에 저장
2 inputSusic = input("수식을 입력하세요 : ")
3 splitedPlus = inputSusic.split("+")
4 splitDone = []
5
6 # for문을 이용해 +를 기준으로 나뉜 수식 조각들을
7 # 다시 -를 기준으로 split하여 splitDone으로 저장
8 for a in splitedPlus :
9     for b in a.split("-") :
10         splitDone.append(b)
11
12 i = 0
13 # 입력된 수식에서 +와 -를 검색하고 이의 위치에 맞게 수식을 계산하여 result에 int 정수형으로 저장
14 result = int(splitDone[0])
15 for c in inputSusic :
16     if c == "+" :
17         i = i + 1
18         result = result + int(splitDone[i])
19     elif c == "-":
20         i = i + 1
21         result = result - int(splitDone[i])
22     else:
23         pass
24
25 print("결과는 %d"%val)
```



C:\Windows\system32\cmd.exe

수식을 입력하세요 : 12+34-5
결과는 41
계속하려면 아무 키나 누르십시오 . . .

<https://github.com/yymin1022/CAU-Computational-Thinking-and-Problem-Solving-Assignment2/blob/master/문제 7.py>

수식을 입력 받고 이를 +와 -에 따라 split 한 뒤 순서에 맞게 수식을 계산하는 코드입니다.

과제를 마치며...

이번 과제를 수행하며 본인의 프로그래밍 과정을 되돌아보는 계기가 되었습니다. Android 애플리케이션, 카카오톡 php 챗봇, 아두이노, Android OS 등 여러가지 주제의 프로그래밍을 진행해본 경험은 있으나 알고리즘 부분에서 취약하여 본 과제를 수행하며 막히는 부분이 많았습니다.

앞으로는 프로그래밍을 진행하며 정말 이것이 최적의 알고리즘인지, 다른 알고리즘에는 무엇이 있는지에 대해 깊게 생각하며 프로그래밍을 진행하고자 노력할 것입니다.