Algorithm Week3

15기 정규세션

TOBIG'S 14기 김상현

Contents

Unit 01 | 1주차 문제 리뷰

Unit 01 | 1주차 문제 리뷰

15기 정규세션
TOBIG'S 14기 김상현

Problem 1. 제곱하기

정수 x를 입력 받은 후 x의 제곱을 출력하는 함수를 정의해서 문제 풀기!

입력1: 3　　출력1: 9
입력2: 0　　출력2: 0
입력3: 11　　출력3: 121

```
1    def square(x):
2        return x**2
3
4    x = int(input())
5
6    print(square(x))
```

Unit 01 | 1주차 문제 리뷰

15기 정규세션
TOBIG'S 14기 김상현

Problem 2. 재윤이의 키보드

투빅스 알고리즘 과제를 풀면서 프로그래밍의 짜릿함을 느끼게 된 재윤이는 즐거운 프로그래밍을 위해 캐럿마켓에서 키보드를 구입했는데,

이게 무슨 일인가. 가볍게 키를 한 번 눌렀음에도 해당 키가 여러 번 입력되는 것이다. 'import'를 입력하려고 했으나 'iiiiiiiimmpppppppppppoooooorrrrrttttttttt'가 되는 것이다.

Whyrano.. 상심에 빠진 재윤이를 위해 해당 현상을 해결해보자!

```python
def solution(arr):
    answer = []
    for i in arr:
        if answer[-1:] == [i]: continue
        answer.append(i)
    return answer

code = input()

print(''.join(solution(code)))
```

Unit 01 | 1주차 문제 리뷰

15기 정규세션
TOBIG'S 14기 김상현

Problem 3. 1이 좋아

투빅이가 좋아하는 숫자는 1이다. 코로나로 심심한 투빅이는 어떤 규칙을 따라가면 특정 숫자가 1이 되는 것을 발견한다. 이를 투빅스 규칙이라 한다. 규칙은 다음과 같다.

양의정수n을입력받았을때($1 \leq n \leq 2^{68}$),

1. n이 짝수라면 2로 나눈다.
2. n이 홀수라면 3을 곱하고 1을 더한다.
3. 1이 될 때까지 반복한다.

투빅스 규칙을 몇 번 반복하면
1이 되는지 알아보는 프로그램을 구현해보자.

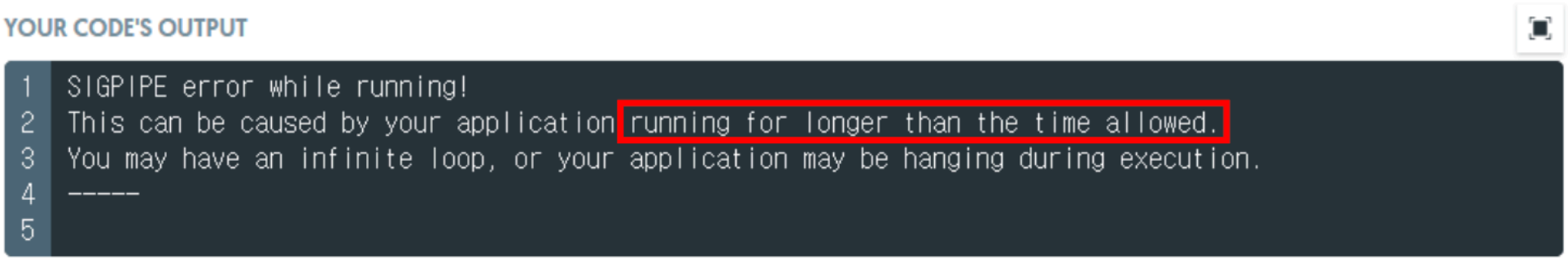
```python
def collatz(num):
    count = 0
    while num > 1:
        if count > 500:
            return -1
        num = num*3 + 1 if num%2 else num/2
        count += 1
    return count


a = int(input())
print(collatz(a))
```

Unit 02 | 시간 복잡도

Unit 02 | 시간 복잡도

15기 정규세션
TOBIG'S 14기 김상현

알고리즘 문제를 풀다보면 시간 초과가 발생할 수 있다!



내가 작성한 코드가 '제한 시간'을 초과하면 발생하는 문제

이를 해결하기 위해 시간복잡도(time complexity) 개념을 짚고 가자!

Unit 02 | 시간 복잡도

시간 복잡도(time complexity)란?

내가 짠 알고리즘이 문제를 해결하기 위해 코드 내에서 얼만큼의 연산이 수행되는가에 대한 지표!
대표적으로 빅오 표기법(big-O notation)이 있다.

빅오 표기법(big-O notation)

가장 영향력 있는 항을 기준으로 표기하여 시간 복잡도를 표현한다. 예) $n^2+2n+1 \Rightarrow O(n^2)$

Unit 02 | 시간 복잡도

15기 정규세션
TOBIG'S 14기 김상현

O(1)

Input 값에 상관없이
항상 고정된 연산을 수행

```
num = int(input())
print(num**2)
```

O(n)

Input값 만큼의
연산을 수행 (선형)

```
num = int(input())
for i in range(num):
    print(i)
```

$O(n^2)$

Input값을 두 번 곱한 만큼의
연산을 수행 (num * num)

```
num = int(input())
for i in range(num):
    for j in range(num):
        print(i+j)
```

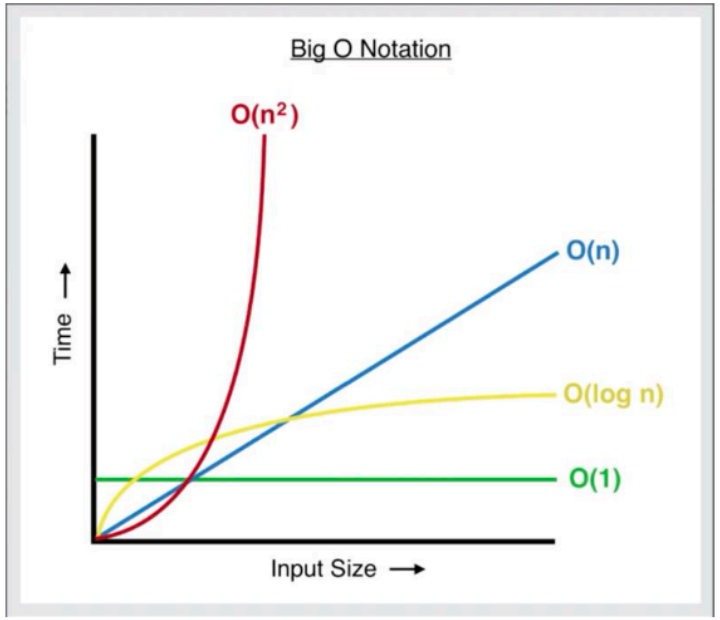
$O(n^3)$

Input값을 세 번 곱한 만큼의
연산을 수행 (num * num * num)

```
num = int(input())
for i in range(num):
    for j in range(num):
        for k in range(num):
            print(k*(i+j))
```

가장 간단하게는 중첩 반복문의 개수로 표기할 수 있다.

Unit 02 | 시간 복잡도



빅오의 차수(N)가 클 수록 시간 복잡도가 수직에 가깝게 증가한다.

$y = x$, $y = x^2$, $y = x^3$ 그래프들..

Unit 02 | 시간 복잡도

15기 정규세션
TOBIG'S 14기 김상현

YOUR CODE'S OUTPUT

```
1  SIGPIPE error while running!
2  This can be caused by your application running for longer than the time allowed.
3  You may have an infinite loop, or your application may be hanging during execution.
4  -----
5
```

시간 초과에 걸렸다면?
가장 먼저 반복문의 개수를 줄여보자!

Unit 03 | 완전 탐색

Unit 03 | 완전 탐색

15기 알고리즘 커리큘럼

1주차	OT & 알고리즘 기초
3주차	완전탐색
5주차	동적계획
7주차	분할정복
9주차	탐욕 알고리즘

Unit 03 | 완전 탐색

완전 탐색(exhaustive search)이란?

모든 경우의 수를 탐색하는 방법!

ex) 4자리 숫자로 된 암호를 찾기 위해 0000 부터 9999까지 모두 입력해보기

Unit 03 | 완전 탐색

장점

- 모든 경우의 수를 탐색하므로 결국 정답을 찾는다..!
- 구현이 쉬운 경우가 많다.

단점

- 경우의 수가 많을 수록 탐색 시간 증가
- 미미르에서 시간 초과..

Unit 03 | 완전 탐색

그럼에도 완전탐색은 효율적인 알고리즘을 짜기 위한 근간이 되어주고,

코딩 테스트에서도 경우의 수를 제한하는 방식 등으로 완전 탐색 문제를 출제하곤 한다.

즉, 우리는 풀 줄 알아야 한다!

Unit 03 | 완전 탐색

[예시]

1 ~ N개의 숫자 중 합이 N이 되는 순서쌍 찾기

N = 10 일 경우

정답 : (1, 9), (2, 8), (3, 7), (4, 6), (5, 5), (6, 4), (7, 3), (8, 2), (9, 1)

Unit 03 | 완전 탐색

N = 10000일 경우, 2중 for문 : $O(n^2)$

```python
start = time.time()

N = 10000

answer = []
for i in range(1, N):
    for j in range(1, N):
        if i+j == N:
            answer.append((i, j))

print("time :", time.time() - start)
```

```
time : 8.491055965423584
```

Unit 03 | 완전 탐색

N = 10000일 경우, 단일 for문 : O(n)

```
1  start = time.time()
2
3  N = 10000
4
5  answer = []
6  for i in range(N):
7      answer.append((i, N-i))
8
9  print("time :", time.time() - start)
```

```
time : 0.0027730464935302734
```

약 4000배 차이!

Unit 04 | 3주차 과제 소개

Unit 04 | 3주차 과제 소개

Problem 1. 엿팔아요

준영이는 엿을 파는 엿장수이다.
매일 아침 인사동에서 엿을 뽑아 팔면서 쏠쏠한 돈 맛을 보고 있다.

준영이의 엿은 뽑은 당일에만 판매할 수 있고, 기계에서 뽑혀져 나오는 엿가락의 길이는 일정하지 않다. 그래서 판매를 위해선 고물상에 있는 엿가위를 빌려 엿을 모두 일정한 길이로 맞춰주어야 한다. 그런데 고물상 주인이 꼰대이므로 한 번 가위질을 할 때마다 돈을 지불해야 한다.

엿을 한 번 자를 때마다 C원이 든다. 잘린 엿은 단위길이당 F원에 판매한다.
즉 자르는 비용을 제외하면 길이가 L인 엿이 N개라면 총 NLF원을 벌 수 있는 것이다. 인사동 인싸 준영이의 장사를 도와주어 오늘 준영이가 벌 수 있는 최대 금액을 구해주자!

Unit 04 | 3주차 과제 소개

Problem 1. 엿팔아요

첫째 줄에 그 날 뽑은 엿가락의 개수 N, 엿을 자를 때 드는 비용 C, 단위길이당 가격 F가 주어진다.

둘째 줄부터 엿 기계에서 뽑혀진 엿의 길이가 한 줄에 하나씩 주어진다. N은 1,000 이하의 자연수, F는 10,000 이하의 자연수, L도 10,000 이하의 자연수이다.

준영이가 벌 수 있는 최대 금액을 출력하자!

입력	출력
3 1 10 26 103 59	1770

Unit 04 | 3주차 과제 소개

Problem 2. IP 채굴기

열심히 엿을 팔아 억만장자가 된 준영이는 돈 씀씀이가 바뀌었다.
필요한 물건이 있으면 필요 이상으로 대량 구매를 하는 습관이 생긴 것이다.

하루는 서버 구축을 위해 IP 주소를 구매해야 했는데, IP주소마저 대량 구매하여 많은 양의 IP주소가 남게되었다.

준영이는 넓은 마음으로 남은 IP주소를 투빅이들에게 나눠주기 위해 16진수를 조합하여 IP주소를 채굴해 쓸 수 있는 문자판을 제작하였다.

다음 문자판에서 채굴 할 수 있는 총 IP주소 개수를 구해보자!

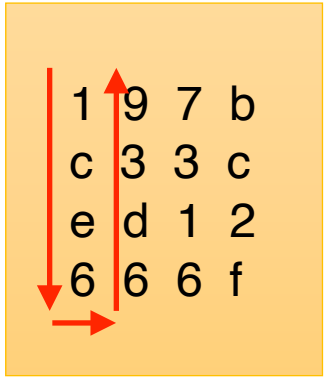
Unit 04 | 3주차 과제 소개

Problem 2. IP 채굴기

문자판은 4 x 4 크기로 이루어져있고, 각 칸에는 16진수(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f)가 적혀있다.

IP를 채굴하는 방법은 다음과 같다.
해당 문자판 임의의 위치에서 시작하여 상, 하, 좌, 우로 각각 움직이면서 8자리의 16진수를 만든다. 이때 한번 이동했던 위치로 다시 돌아갈 수 있다.

만약 0f0f0f0f를 만들었다면 2진수로 변경한
00001111.00001111.00001111.00001111이 IP주소가 되며,
15.15.15.15를 채굴한 것이다!



1ce66d39

Unit 04 | 3주차 과제 소개

Problem 2. IP 채굴기

입력 1 1 1 1
1 1 1 1
1 1 1 1
2 1 1 1

출력 31

채굴 가능한 IP는 다음과 같다.

['11111111', '11111112', '11111212', '11111211', '11112121', '11112111', '11111121', '11121212', '11121211', '11121112', '11121111', '11212121', '11212111', '11211121', '11211111', '12121212', '12121211', '12121112', '12121111', '12111212', '12111211', '12111112', '12111111', '21212121', '21212111', '21211121', '21211111', '21112121', '21112111', '21111121', '21111111']

Unit 04 | 3주차 과제 소개

Problem 3. 괴도 예은

투빅스 보석방에는 N개의 다이아몬드가 있다. 보석방 주인 준영이는 돈이 엄청나게 많아서 다이아몬드를 훔쳐가는 것에 관대하다. 따라서 하루에 도난당한 다이아몬드 가격의 합이 M원을 넘지 않는다면 경찰에 신고하지 않고 넘어간다.

이를 악용하여 괴도 예은은 N개의 다이아몬드 중 3개를 훔쳐오라는 지령을 받아 투빅스 보석방에서 3개의 다이아몬드를 훔치려 한다. 경찰서에서 콩밥을 먹기 싫고 돈은 많이 벌고 싶은 욕심 때문에 훔친 다이아몬드 3개 가격의 합이 최대가 되면서 M원을 넘으면 안된다.

N개 다이아몬드의 가격들이 각각 주어졌을 때, M원을 넘지 않으면서 M원에 최대한 가까운 다이아몬드 3개 가격의 합을 구해보자.

Unit 04 | 3주차 과제 소개

Problem 3. 괴도 예은

입력
첫째 줄에 다이아몬드의 개수 N(3<= N <= 100)과 M(10 <= M <= 300000)이 주어진다. 둘째 줄에는 다이아몬드 각각의 가격이 주어진다. 다이아몬드 각각의 가격은 100,000원을 넘지 않는다.

출력
첫째 줄에 M을 넘지 않으면서 M에 최대한 가까운 다이아몬드 3개 가격의 합을 출력한다.

입력	출력
5 21 5 6 7 8 9	21

Unit 04 | 3주차 과제 소개

코스 코드

42a4b8acf7

코스 링크

https://class.mimir.io/courses/42a4b8acf7/registrations/new

둘 중 아무 방식으로 참여 하시면 됩니다~!

들어주셔서 감사합니다.