알고리즘 Week 3

20기 정규세션

TOBIG'S 19기 한진모

Contents

Unit 01 | 1주차 과제 리뷰

20기 정규세션
TOBIG'S 19기 한진모

2438. 별 찍기 - 1

https://www.acmicpc.net/problem/2438

첫째 줄에는 별 1개, 둘째 줄에는 별 2개, N번째 줄에는 별 N개를 찍도록 프로그램을 작성하시오.

[입력]

첫째 줄에 N(1 ≤ N ≤ 100)이 주어진다.

[출력]

첫째 줄부터 N번째 줄까지 차례대로 별을 출력한다.

[입력 예시]

5

[출력 예시]

```
*
**
***
****
*****
```

20기 정규세션
TOBIG'S 19기 한진모

문제 1. 별 찍기

```python
N = int(input())

#첫째 줄부터 N번째 줄까지
for i in range(N):
    #i번째 줄에선 별을 (i+1)개 출력한다.
    print('*' * (i+1) )
```

20기 정규세션
TOBIG'S 19기 한진모

1009. 분산처리 https://www.acmicpc.net/problem/1009

[입력]

입력의 첫 줄에는 테스트 케이스의 개수 T가 주어진다.
그 다음 줄부터 각각의 테스트 케이스에 대해 정수 a와 b가 주어진다. (1 ≤ a < 100, 1 ≤ b < 1,000,000)

[출력]

각 테스트 케이스에 대해 마지막 데이터가 처리되는 컴퓨터의 번호를 출력한다.

[입력 예시]

```
5
1 6
3 7
6 2
7 100
9 635
```

[출력 예시]

```
1
7
6
1
9
```

20기 정규세션
TOBIG'S 19기 한진모

문제 2. 분산 처리

```python
N = int(input())

#a의 b제곱을 10으로 나눈 나머지 단순 계산 시 수가 커지면 시간초과.
#중요한 정보는 일의 자리뿐이며, 반복주기의 최소공배수는 4이다.
residue_mtx = [[10,10,10,10],#10번 컴퓨터이므로
               [1,1,1,1],
               [2,4,8,6],
               [3,9,7,1],
               [4,6,4,6],
               [5,5,5,5],
               [6,6,6,6],
               [7,9,3,1],
               [8,4,2,6],
               [9,1,9,1]]

for i in range(N):
    a, b = map(int, input().split())
    print(residue_mtx[a%10][(b-1)%4])
```

중요한 정보는 일의 자리뿐임을 이용해,
미리 나머지의 행렬을 갖고 있는 것이 핵심.

20기 정규세션
TOBIG'S 19기 한진모

2563. 색종이 https://www.acmicpc.net/problem/2563

[입력]

첫째 줄에 색종이의 수, 둘째 줄부터 색종이를 붙인 위치가 주어진다. 색종이를 붙인 위치는 두 개의 자연수로 주어지는데 첫 번째 자연수는 색종이의 왼쪽 변과 도화지의 왼쪽 변 사이의 거리이고, 두 번째 자연수는 색종이의 아래쪽 변과 도화지의 아래쪽 변 사이의 거리이다. 색종이의 수는 100 이하이며, 색종이가 도화지 밖으로 나가는 경우는 없다

[출력]

첫째 줄에 색종이가 붙은 검은 영역의 넓이를 출력한다.

[입력 예시]

```
3
3 7
15 7
5 2
```

[출력 예시]

```
260
```

20기 정규세션
TOBIG'S 19기 한진모

문제 3. 색종이

```python
N = int(input())
canvas = [[0 for _ in range(100)] for _ in range(100)]

for i in range(N):
    a, b = map(int, input().split())
    for j in range(10):
        for k in range(10):
            canvas[a+j][b+k] = 1

totalSum = 0
for row in canvas:
    totalSum += sum(row)

print(totalSum)
```

20기 정규세션
TOBIG'S 19기 한진모

1475. 방 번호 https://www.acmicpc.net/problem/1475

[입력]

첫째 줄에 다솜이의 방 번호 N이 주어진다. N은 1,000,000보다 작거나 같은 자연수이다.

[출력]

첫째 줄에 필요한 세트의 개수를 출력한다.

[입력 예시 1]

9999

[출력 예시 1]

2

[입력 예시 2]

122

[출력 예시 2]

2

[입력 예시 3]

12635

[출력 예시 3]

1

20기 정규세션
TOBIG'S 19기 한진모

문제 4. 방 번호

```python
import math

N_list = list(input())
N_count = {}

for i in N_list:
    #6과 9를 모두 같은 수로 취급합니다.
    if i == '9':
        i = '6'
    #각 숫자가 몇 번 등장했는지 딕셔너리를 만들겠습니다.
    #참고: 파이썬의 Counter 모듈을 사용하면 훨씬 쉽게 만들 수 있습니다.
    try:
        N_count[i] += 1
    except:
        N_count[i] = 1

#6과 9는 서로 호환되므로, 2로 나눈 뒤 올림한 횟수만큼 세트가 필요합니다.
try:
    N_count['6'] = math.ceil(N_count['6'] / 2)
except:
    pass

print(max(N_count.values()))
```

6과 9의 처리가 핵심.

20기 정규세션
TOBIG'S 19기 한진모

2503. 숫자 야구 https://www.acmicpc.net/problem/2503

민혁이와 영수는 숫자 야구 게임을 진행중이다.

민혁이가 영수에게 어떤 수들을 물어보았는지, 그리고 각각의 물음에 영수가 어떤 대답을 했는지가 입력으로 주어진다. 이 입력을 바탕으로 여러분은 영수가 생각하고 있을 가능성이 있는 수가 총 몇 개인지를 알아맞혀야 한다.

20기 정규세션
TOBIG'S 19기 한진모

문제 5. 숫자 야구

```python
from itertools import permutations
N = int(input())

digits = ['1', '2', '3', '4', '5', '6', '7', '8', '9']
num = list(permutations(digits, 3))

for _ in range(N):
    query, scount, bcount = map(int, input().split())
    query = list(str(query))
    removed = 0
    for i in range(len(num)):
        strike = ball = 0
        i -= removed # num[0] 부터 시작
        for j in range(3):
            if num[i][j] == query[j]:
                strike += 1
            elif query[j] in num[i]:
                ball += 1

        if (strike != scount) or (ball != bcount):
            num.remove(num[i])
            removed += 1

print(len(num))
```

가능한 모든 조합으로부터,
Query가 한 번씩 들어올 때마다 탐색 범위를 좁혀나간다.

Unit 02 | 시간 복잡도

Unit 02 | 시간 복잡도

'효율적인' 알고리즘을 위한 고민

- 해답을 찾는 것도 중요하지만, '효율적인' 방법으로 문제를 풀었는지도 중요
- '효율성'을 고려한다 = 시간 복잡도를 고려한다

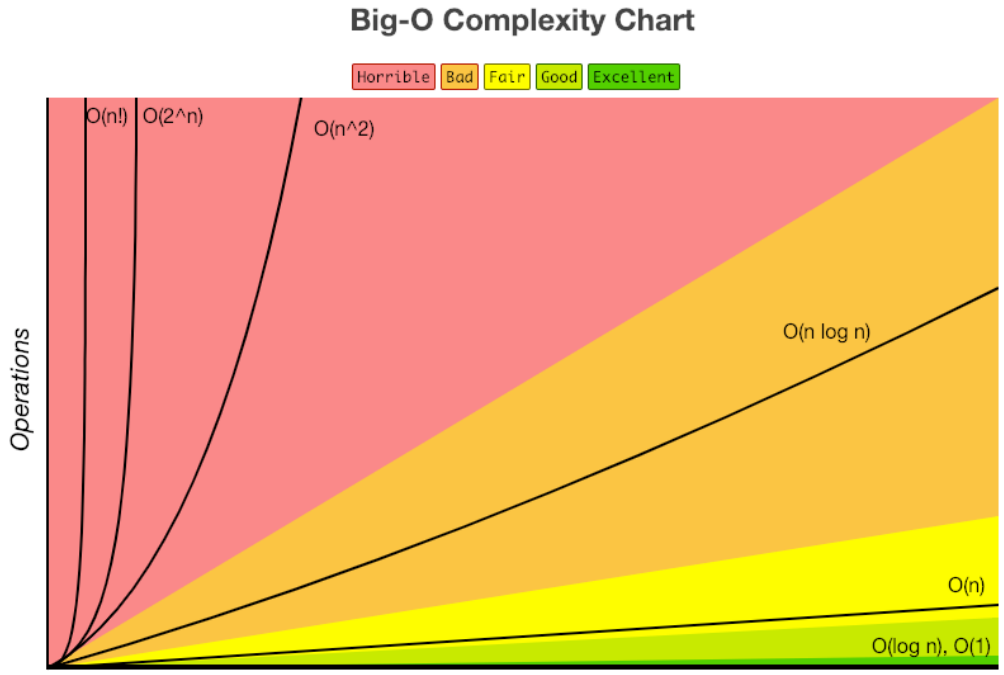
시간 복잡도 (Time Complexity)

- 알고리즘을 풀어낼 때 프로세스가 수행해야 하는 연산을 수치화한 것
- 주로 Big-O 표기법을 이용해 시간 복잡도를 표현

Unit 02 | 시간 복잡도

Big-O 표기법

- 가장 영향력 있는 항을 기준으로 표기
  - 예 : $n^2 + 2n + 1 \quad \rightarrow \quad O(n^2)$



O(1)
- 입력에 관계없이 동일한 시간 복잡도 (고정된 연산)
- def hello_world():
  print("hello, world!")

O(n)
- 입력값이 증가하면 처리 시간이 선형적으로 증가
- def print_each(li):
  for item in li:
  print(item)

O(n^2)
- 반복문이 두 번 사용된 경우
- def print_each_n_times(li):
  for n in li:
  for m in li:
  print(n,m)

Unit 02 | 시간 복잡도

0x00 시간, 공간복잡도

- N의 크기에 따른 허용 시간복잡도는 대략 아래와 같습니다.(맹신하지는 마세요.)

N의 크기	허용 시간복잡도
$N \leq 11$	$O(N!)$
$N \leq 20$	$O(2^N)$
$N \leq 100$	$O(N^4)$
$N \leq 500$	$O(N^3)$
$N \leq 3{,}000$	$O(N^2 lgN)$
$N \leq 5{,}000$	$O(N^2)$
$N \leq 1{,}000{,}000$	$O(NlgN)$
$N \leq 10{,}000{,}000$	$O(N)$
그 이상	$O(lgN), O(1)$

15

출처: https://blog.encrypted.gg/724

Unit 03 | 완전탐색

Unit 03 | 완전탐색

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

완전탐색 (Exhaustive Search, Brute-Force)

- 컴퓨터의 빠른 계산 능력을 이용하여 가능한 경우의 수를 일일이 나열하면서 답을 찾는 방법
  - 예 : 4자리 숫자로 된 암호를 찾기 위해 0000부터 9999까지 모두 입력해보기

- **종류**
  - Brute Force 기법 - 반복문 / 조건문 이용
  - 백트래킹 (Backtracking) - 가지치기, 분할정복 이용 (7주차)
  - 순열 (Permutation) - 서로 다른 N개를 일렬로 나열
  - 비트 마스크 (Bit Mask) - 이진수 이용
  - 재귀함수
  - BFS/DFS - 그래프 자료 구조에서 모든 정점을 탐색

완전탐색 (Exhaustive Search, Brute-Force)

- **장점**
  - 직관적이므로 이해하기 쉽고, 구현도 간단하다.
  - 즉, 문제의 정확한 결과값을 얻어낼 수 있는 가장 확실하며 기초적인 방법

- **단점**
  - 시간 초과 문제

**BUT, 완전탐색은 효율적인 알고리즘을 짜는 근간이면서,
코딩 테스트에서 경우의 수를 제한하는 방식 등으로 출제된다!**

Unit 04 | 3주차 과제 소개

20기 정규세션
TOBIG'S 19기 한진모

문제 1. DNA(https://www.acmicpc.net/problem/1969)

[입력 조건]
DNA의 수 N과 문자열 길이 M이 주어진다. N은 1000보다 작거나 같은 자연수이고, M은 50보다 작거나 같은 자연수이다.

[출력 조건]
첫째 줄에 Hamming Distance의 합(차이 나는 각 염기서열간 거리의 합)이 가장 작은 DNA를 출력하고, 둘째 줄에 Hamming Distance의 합을 출력하시오. 해당 DNA가 여러 개 있을 땐 사전순으로 가장 앞서는 것을 출력하시오.

[입력 예시]
5 8 #(DNA의 수, 문자열의 길이)
TATGATAC
TAAGCTAC
AAAGATCC
TGAGATAC
TAAGATGT

[출력 예시]
TAAGATAC
7

문제 2. 오목

(https://www.acmicpc.net/problem/2615)

[입력 조건]
흑돌이 1, 백돌이 2, 두어지지 않은 칸이 0으로 주어진 19x19 바둑판이 주어진다.

[출력 조건]
첫 줄에 흑 승리 시 1, 백 승리 시 2, 아직 승부가 결정되지 않았을 시 0을 출력한다(육목 인정 X).
흑/백 승리 시 연속된 바둑돌 중 좌상단 끝에 있는 바둑돌의 좌표를 아래와 같이 출력한다.

[입력 예시]

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 2 0 0 2 2 2 1 0 0 0 0 0 0 0 0 0 0
0 0 1 2 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
0 0 0 1 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 2 2 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 2 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

[출력 예시]
1 #(흑 승리)
3 2 #(3행 2열)

20기 정규세션
TOBIG'S 19기 한진모

문제 3. 치킨 배달

(https://www.acmicpc.net/problem/15686)

[입력 조건]
첫째 줄에 정사각형 도시의 한 변의 길이 N과 남겨두어야 할 치킨집의 수 M이 주어진다.
둘째 줄부터 N개의 줄에는 도시의 정보가 주어진다. 0은 빈 칸, 1은 집, 2는 치킨집이다.

[출력 조건]
도시에서 M개의 치킨집만을 남겨야 한다. 어떻게 남겨야 각 집과 치킨집의 격자 거리(맨하탄 거리)를 최소화할 수 있는가? 최솟값을 출력하시오.

[입력 예시]

```
5 3
0 0 1 0 0
0 0 2 0 1
0 1 2 0 0
0 0 1 0 0
0 0 0 0 2
```

[출력 예시]
5

다시 한번 리마인드

구글 form 및 BOJ가입 신청 꼭! 진행해주세요

- 구글 form: https://forms.gle/TE14gpNnauoygCCr9
- BOJ그룹 가입: https://www.acmicpc.net/login?next=%2Fgroup%2Fjoin%2F18403

질문/힌트/모든 문의는 19기 김은지/ 위성진/ 이동준/ 임서영 최다희/ 표지원/ 한진모 에게

언제든 자유롭게 해주세요!