Computing Foundations for Data Science

HW #14

제출기한: 2022/5/10 14:00PM

주의사항

- 코드를 Jupyter Notebook 에서 작성하였더라도 python 파일(.py)로 제출할 것.
- 함수가 의도한 값을 return 하는지를 확인할 것. (print 와 혼동하지 말 것)
- 파일명은 P1.py ~ P5.py 를 유지하고, 해당 파일들을 HW14_학번_이름.zip 으로 압축하여 제출할 것. 예를 들면 학번이 2020-12345 이고, 이름이 Keondo Park 이라면 **HW14_2020_12345_KeondoPark.zip** 으로 압축하여 제출.
    - 압축 시 **반드시 zip** 으로 할 것. (egg, tar, gz, rar 등은 채점 대상 제외)
- 각 파일들은 문제를 해결하기 위한 함수만 있어야 하며 불필요한 출력이 있을 시 불이익 받을 수 있음.
    - 에시) P1.py 에는 P1 함수만 있어야 함.
    - 테스트를 위한 P1 함수 호출이 있을 시 불이익이 있을 수 있음.
    - 제출 시에 print 등 debug/test 를 위한 코드는 지우고 제출할 것
- 예시로 제시한 입력 값 외에도 조교가 임의로 생성한 입력 값으로도 코드가 잘 실행되는지 테스트할 예정.
- 뼈대 코드의 함수 이름 및 매개변수(parameter)는 **변경하지 말 것**.
- 채점은 프로그램에 의해 기계적으로 처리되므로 위 사항을 지키지 않은 경우 누락되거나 불이익을 받을 수 있음.
- 문제에서 불명확하거나 모호한 부분은 slack 을 활용하여 질문할 것.
- **늦은 제출은 받지 않음.**
- 표절 검사를 수행하여 발각될 경우 성적 **F** 부여.
- 이번 과제는 bitwise operation 에 대한 과제로, bitwise operation 을 이용해서 문제를 해결하는 것을 권장함. (채점은 return 값으로만 진행)
    - Bitwise operation: &, ^, >>, << 등

P1.

주어진 **양의 정수 n** 을 이진수로 나타냈을 때 교차 비트로 이루어져 있는지를 확인하여 **bool 값**으로 return 하는 P1 함수를 구현하여라. 교차 비트로 이루어져 있다면 True, 이루어져 있지 않다면 False 를 return 한다.

- 교차 비트란 0101... 혹은 1010...으로 인접한 비트가 항상 다른 비트인 것을 의미한다.
- 양 옆에 인접 비트가 없을 때는 True 를 반환한다. (ex. 1)
- 1 과 0 의 개수가 다르면서 교차인 경우(10101 과 같은 경우)에도 교차 비트로 이루어진 것으로 판단하여 True 를 반환한다.

예시 1)

>>> P1(8)

False

설명: 8 을 이진수로 나타내면 1000 이기 때문에 교차 비트로 이루어져 있지 않아서 False 를 return 한다.

예시 2)

>>> P1(10)

True

설명: 10 을 이진수로 나타내면 1010 이기 때문에 교차 비트로 이루어져 있어 True 를 return 한다.

예시 3)

>>> P1(1)

True

설명: 1 을 이진수로 나타내면 1 이기 때문에 인접 비트가 없으므로 True 를 return 한다.

예시 4)

>>> P1(4294967295)

False

설명: 4294967295 를 이진수로 나타내면 11111111111111111111111111111111 이기 때문에 교차 비트로 이루어 있지 않아서 False 를 return 한다.

P2.

0 이상 정수 num 을 입력으로 받는다. Num 을 32bit 로 나타낸 다음 (반드시 32bit 로 나타내어야 한다. 앞자리 수가 비더라도 0 으로 채워서 32 자리를 맞춰야 함), bit 를 거꾸로 뒤집었을 때의 값을 0 이상의 10 진수 정수로 return 하는 P2 함수를 구현하시오.

예시 1)

>>>P2(43261596)

964176192

설명: 43261596 은 32bit 로 나타냈을 때, 00000010100101000001111010011100 이고, 이것을 뒤집으면 00111001011110000010100101000000 가 되고, 이 값은 964176192 이다

예시 2)

>>> P2(4294967293)

3221225471

설명: 4294967293 은 32bit 로 나타냈을 때, 11111111111111111111111111111101 이고, 이것을 뒤집으면 10111111111111111111111111111111 가 되고, 이 값은 3221225471 이다.

예시 3)

>>> P2(123456789)

2830359264

설명: 123456789 은 32bit 로 나타냈을 때, 00000111010110111100110100010101 이고, 이것을 뒤집으면 10101000101100111101101011100000 가 되고, 이 값은 2830359264 이다.

P3.

정수로 이루어진 **리스트 nums** 를 인자로 받는다. nums 의 성분 중, 두 개의 정수는 한 개씩만 있고, 나머지 정수들은 두 개씩 있다. 한 개씩 있는 두 개의 정수를 **set** 으로 return 하는 P3 함수를 구현하시오.

구현 방법에는 여러 가지가 있을 수 있으나, bit operation 을 이용해서 구현해보자. Time complexity O(n), 입력으로 들어가는 nums 를 제외한 space complexity O(1)에 구현할 수 있다. (채점은 구현에 상관없이 return 값으로만 채점할 것이다.)

예시 1)

>>> P3([1,2,1,3,2,5])

{3, 5}

설명: 3 과 5 는 한 개씩 있고, 나머지 정수들은 두 개씩 있다. set 으로 return 하기 때문에 순서는 상관없다.

예시 2)

>>> P3([-1,0])

{-1, 0}

예시 3)

>>> P3([1,2,3,4,5,6,7,8,9,1,2,3,4,5,6,7])

{8, 9}

P4.

0 이상 정수 num 을 입력으로 받는다. bit 로 표현했을 때, '1'의 개수를 return 하는 함수를 구현하시오. (num 은 32bit 를 넘지 않는다.)

예시 1)

>>> P4(16)

1

예시 2)

>>> P4(7)

3

예시 3)

>>> P4(1234567)

11