**Computing Foundations for Data Science**

**HW #8**

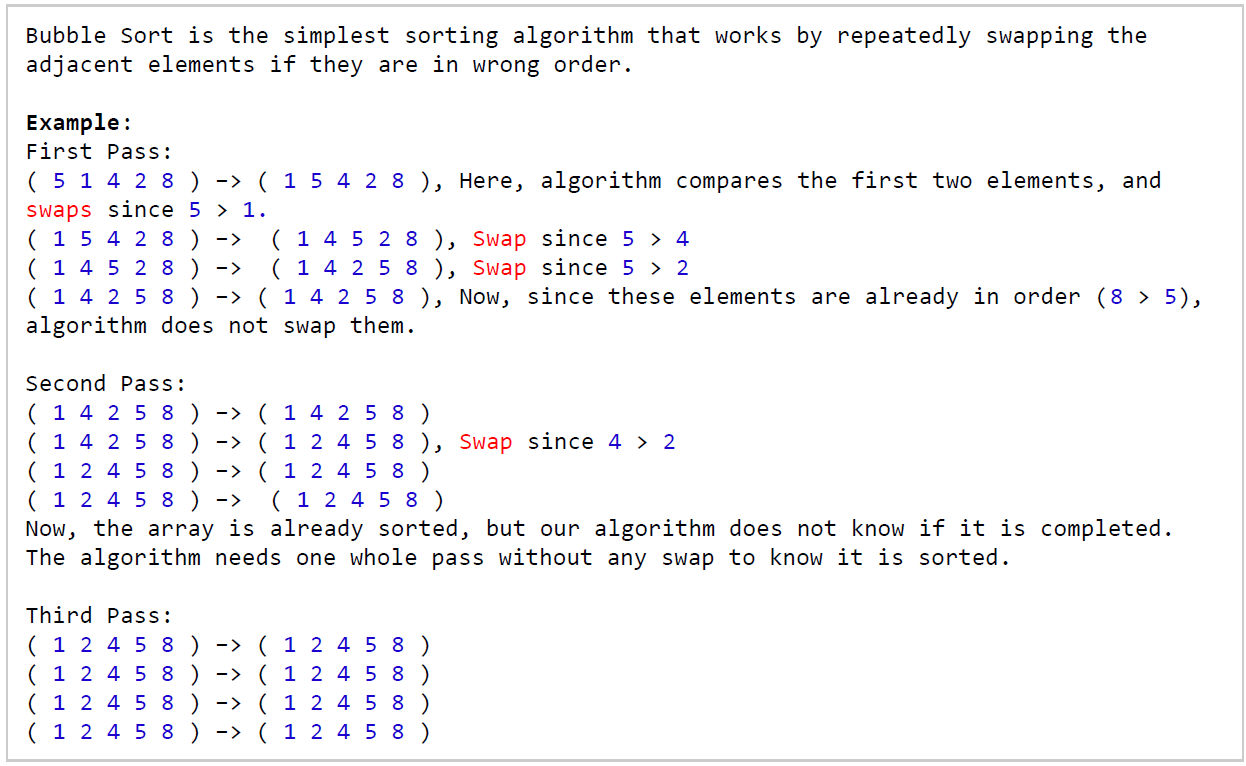
**제출기한: 2022/4/5 14:00PM**

**주의사항**

* 코드를 Jupyter Notebook에서 작성하였더라도 python 파일(.py)로 제출할 것.
* 함수가 의도한 값을 return하는지를 확인할 것. (print와 혼동하지 말 것)
* 파일명은 P1.py ~ P10.py를 유지하고, 해당 파일들을 HW8\_학번\_이름.zip으로 압축하여 제출할 것. 예를 들면 학번이 2020-12345이고, 이름이 Keondo Park이라면 **HW8\_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.py에는 P1 함수만 있어야 함.
  + 테스트를 위한 P1함수 호출이 있을 시 불이익이 있을 수 있음.
  + 제출 시에 print 등 debug/test를 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 slack을 활용하여 질문할 것.
* **늦은 제출은 받지 않음.**
* 표절 검사를 수행하여 발각될 경우 성적 **F** 부여.

**문제 1.**

버블 정렬(bubble sort)은 인접한 두 원소의 순서가 바뀌어 있을 때 (앞의 원소가 뒤의 원소보다 클 때) 서로 swap하면서 정렬하는 알고리즘이다.



정수로 이루어진 리스트(list)를 변수(argument)로 받아 버블 정렬할 때, swap이 일어나는 횟수를 리턴하는 함수를 작성하시오.

**예시1)**

>>> P1([5, 1, 4, 2, 8])

4

**예시2)**

>>> P1([-1, -1, -1, -1, -1])

0

**예시3)**

>>> P1([6, 5, 4, 3, 2, 1])

15

**예시4)**

>>> P1([1, 5, 2, 3, 6, 6, 1, 2, 3, 4, 21, 1, 11, -1])

40

**문제 2.**

두 문자열(string) Order와 S가 있다. Order의 모든 문자는 중복이 없고 어떤 규칙에 의해 이미 정렬되어 있다. S를 다음 규칙에 따라 수정하여 Correct\_S라는 새로운 문자열을 만들려고 한다.

**규칙1>** 만약 S에 Order에 포함된 문자가 있으면 Order에서 사용된 규칙에 의해 우선 정렬되어야 한다.

**규칙2>** Order에 없는 문자에 대해서는 원래 S의 순서를 따르며, 규칙 1에서 정렬된 문자열 뒤에 나타내야 한다.

**규칙3>** S는 Order의 일부만 포함하거나 아예 포함하지 않는 경우, 중복된 경우도 있다.

Order와 S를 변수(arguments)로 받아 Correct\_S를 리턴하는 함수 **P2(Order, S)**를 작성하시오.

**예시1)**

>>> P2(‘cba', ‘abcd’)

‘cbad’

**예시2)**

>>> P2('cba', 'xyhabcd')

'cbaxyhd'

**예시3)**

>>> P2('cbafg', 'xabcd')

'cbaxd'

**문제 3.**

다음 조건을 만족하는 리스트(list)가 있다.

1. 모든 원소는 알파벳 소문자로 된 문자열(string)이다. (공백 없음)

2. 각 string의 길이는 1 이상 20 이하이다.

이 리스트를 변수(argument)로 받아 다음 조건에 따라 정렬하여 리턴하는 함수를 작성하시오.

1. 길이가 짧은 문자열이 앞에 있어야 한다.

2. 길이가 같다면 사전 순서가 빠른 문자열이 앞에 있어야 한다.

>>>P3(['solve', 'this', 'problem', 'or', 'you', 'will', 'get', 'f'])

['f', 'or', 'get', 'you', 'this', 'will', 'solve', 'problem']

>>>P3(['computing', 'class', 'is', 'so', 'funny', 'haha'])

['is', 'so', 'haha', 'class', 'funny', 'computing']

>>>P3(['making', 'homework', 'is', 'very', 'hard'])

['is', 'hard', 'very', 'making', 'homework']

**문제 4.**

행렬은 각 행을 리스트로 가지는 리스트로 표현할 수 있다.

****

즉, 위 그림과 같은 행렬은 다음과 같이 표현한다.

[[1, 3, 0], [2, 5, 3], [3, 4, 1], [2, 0, 0]]

행렬을 변수(argument)로 받아서 아래 그림과 같이 오른쪽 아래 대각선 방향으로 정렬한 행렬을 리턴하는 함수를 작성하시오.

****

**예시1)**

>>> P4([[1, 3, 0], [2, 5, 3], [3, 4, 1], [2, 0, 0]])

[[1, 3, 0], [0, 1, 3], [0, 2, 5], [2, 3, 4]]

**예시2)**

>>> P4([[1, 6, 2, 4, 6, 2, 4, 7, 0]])

[[1, 6, 2, 4, 6, 2, 4, 7, 0]]

**예시3)**

>>> P4([[1, 5, 20, -1, 3], [-1, 3, 20, 4, -1], [34, 3, 12, 5, -12], [4, 64, 612, 6, 10]])

[[1, 5, -12, -1, 3],

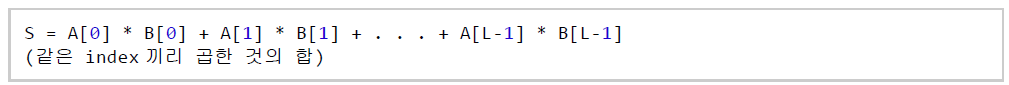
[-1, 3, 5, 4, -1],

[34, 3, 6, 10, 20],

[4, 64, 612, 12, 20]]

**문제 5.**

길이가 L이고, 음이 아닌 정수로 이루어진 두 list A, B가 있을 때, A를 재배열해서 다음 값을 최소로 만들려고 한다.



최소가 되는 S를 return하는 함수 **P5(A,B)**를 구현하시오. (list A, B를 입력으로 받음)

**예시1)**

>>> P5([1,1,1,6,0], [2,7,8,3,1])

18

**예시2)**

>>> P5([1,4,2,6,1,0], [3,5,1,2,4,1])

21

**예시3)**

>>> P5([1,0,1,0,1], [0,1,0,1,0])

0