

Computing Foundations for Data Science

HW 9

제출기한: 2022/04/12 14:00 PM

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 ~ P4.py 를 유지하고, 해당 파일들을 HW9_학번_이름.zip 으로 압축하여 제출할 것. 예를 들면 학번이 2020-12345이고, 이름이 Keondo Park 이라면 **HW9_2020_12345_KeondoPark.zip** 으로 압축하여 제출.
 - 압축 시 **반드시 zip** 으로 할 것. (egg, tar, gz, rar, 등은 0점. 채점 대상 제외)
- 각 파일들은 문제를 해결하기 위한 함수만 있어야 하며 불필요한 출력이 있을 시 불이익 받을 수 있음
 - 예시) P1.py 에는 def P1() 함수만 있어야함.
 - 테스트를 위한 P1() 함수 호출이 있을 시 불이익 이 있을 수 있음
 - 제출시에 print 문은 지워서 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.
- 문제의 instruction 이 불명확하거나 clarification 이 필요할 경우 slack 을 활용하여 질문할 것.
- **늦은 제출은 받지 않음.**
- 표절 검사를 수행하여 발각될 경우 성적 **F** 부여.

문제 1.

'(', ')', '{', '}', '[', ']'으로 구성된 String을 Input으로 받고, 이 String이 아래 규칙에 의해 유효한지를 판단하여 True/False를 return하는 함수를 작성하여라.

(1) 괄호가 열렸다면 같은 타입의 괄호에 의해 닫혀야 한다.

(2) 적절한 순서로 괄호가 배열되어야 한다. 안쪽에서 열린 괄호가 먼저 닫히고, 바깥쪽 괄호가 닫혀야 한다.

예시1)

```
>>> P1('()')
```

```
True
```

예시2)

```
>>> P1('(){}')
```

```
True
```

예시3)

```
>>> P1('([)]')
```

```
False
```

설명: 안쪽에서 열린 '['가 닫히기 전에 '('가 먼저 닫혔으므로 유효하지 않다.

문제 2.

예지는 HS컴퓨팅이라는 주식에 단기투자를 통하여 수익을 내려고 한다. 아래 표는 HS컴퓨팅의 날짜 별 주가이다. 예지는 수익을 내기 위하여 걸리는 최소 며칠을 기다려야 하는지, 즉 최소 일수를 알고싶어 한다.

날짜	10/7	10/8	10/9	10/10	10/11	10/12	10/13	10/14
주가	500	600	700	300	100	400	800	500
일수	1	1	4	2	1	1	0	0

위의 표를 보면

(1) 예지가 10/7에 주식을 산다면 1일 뒤인 10/8에 팔면 이득을 본다.

(2) 예지가 10/9에 주식을 산다면 4일 뒤인 10/13에 팔면 이득을 본다.

(3) 이득을 볼 수 없다면 0으로 작성하면 된다.

위 설명을 참고하여 Input으로 일별 주가(list)를 받고, 이득을 보기 위해 기다려야 하는 최소 일수(list)를 return하는 함수를 작성하여라.

※ Hint: Stack 구조를 활용하여 구현할 수 있다.

예시)

```
>>> P2([500, 600, 700, 300, 100, 400, 800, 500])
```

```
[1, 1, 4, 2, 1, 1, 0, 0]
```

문제 3.

주어진 Singly linked list를 뒤집는 함수를 작성하여라. 함수의 Input은 Singly linked list의 첫번째 Node이며, Return값은 순서가 뒤집힌 Singly linked list의 첫번째 Node이다.

※ Linked list의 각 Node는 linked_list_helper.py 파일에 정의되어 있는 ListNode를 활용하면 된다.

※ linked_list_helper.py 파일의 create_linked_list 함수와 print_linked_list 함수를 활용하여 아래와 같이 함수의 동작을 확인할 수 있다.

※ Space complexity는 $O(1)$ 이어야 합니다.

예시) 새로운 LinkedList 생성, list객체 생성 등 추가 선언은 모두 0점 처리

예시1)

```
>>> l1 = create_linked_list([4,2,1,3])
>>> print_linked_list(P3(l1),[])
[3,1,2,4]
```

예시2)

```
>>> l2 = create_linked_list([-1,5,0,2,3])
>>> print_linked_list(P3(l2),[])
[3,2,0,5,-1]
```

예시3)

```
>>> l3 = create_linked_list([])
>>> print_linked_list(P3(l3),[])
[]
```

문제 4.

0 이상의 십진수 정수를 나타내는 Linked list 두 개가 주어질 때, 두 숫자의 합을 나타내는 Linked list를 만드는 함수를 작성하여라. Input으로는 각 Linked list의 첫번째 Node가 주어지며, Return값 역시 Linked list의 첫번째 Node이다.

예를 들어, 15와 30을 나타내는 Linked list, 즉 (1)->(5), (3)->(0)이 주어진다면, 두 숫자의 합인 45를 나타내는 Linked list (4)->(5)를 만들어 해당 Linked list의 첫번째 Node (head)를 반환하여야 한다.

※ Linked list 형태를 유지하며 계산할 것 (주어진 Linked list를 list로 변환하여 각 원소를 더하고, 이를 다시 Linked list로 만드는 등의 풀이는 허용하지 않음)

※ Linked list의 각 Node는 linked_list_helper.py 파일에 정의되어 있는 ListNode를 활용하면 된다.

※ linked_list_helper.py 파일의 create_linked_list 함수와 print_linked_list 함수를 활용하여 아래와 같이 함수의 동작을 확인할 수 있다.

예시1)

```
>>> l1 = create_linked_list([1,1,1])
>>> l2 = create_linked_list([1,1,1])
>>> print_linked_list(P4(l1, l2),[])
[2,2,2]
```

예시2)

```
>>> l1 = create_linked_list([0])
>>> l2 = create_linked_list([0])
>>> print_linked_list(P4(l1, l2),[])
[0]
```

예시3)

```
>>> l1 = create_linked_list([0])
>>> l2 = create_linked_list([1])
```

```
>>> print_linked_list(P4(l1, l2),[])
```

```
[1]
```

예시4)

```
>>> l1 = create_linked_list([5,5,5])
```

```
>>> l2 = create_linked_list([5,5,5])
```

```
>>> print_linked_list(P4(l1, l2),[])
```

```
[1,1,1,0]
```

예시5)

```
>>> l1 = create_linked_list([5,5,5])
```

```
>>> l2 = create_linked_list([5,5])
```

```
>>> print_linked_list(P4(l1, l2),[])
```

```
[6,1,0]
```