

Computing for Data Science

Project #2

제출기한: 2021.04.12 10:59

문제 1. 경로 단순화

파일 경로를 입력받아 단순화된 경로를 반환하는 함수를 작성하여라. 단순화 법칙은 아래와 같다.

- 입력경로에서 '.'은 현재 폴더를 의미하고, '..'는 상위 폴더를 의미한다. '...'은 파일 또는 폴더명에 포함되는 것으로 취급한다.
- 반환값에서 '.' 또는 '..'는 사용하지 않는다.
- 입력경로에서 파일/폴더는 '/'로 구분되며, '/'는 '/'와 동일하게 취급된다. 반환값에서는 '/'로만 경로구분이 되어야 한다.
- 반환값은 '/'로 시작되어야 하고, 마지막에 붙은 '/' 또는 '/'는 제거한다.
- 시작 폴더는 root 폴더라고 가정한다. 즉 시작 폴더에는 상위 폴더가 없다. 시작 폴더를 반환하는 경우는 '/'를 반환하여라.

예시 1) 마지막의 '/'는 제거하여야 한다.

```
>>> P1('/home/')
```

```
'/home'
```

예시 2) 마지막의 '/'를 제거하고 현재 폴더의 상위 폴더를 반환한다. 그렇지만 시작폴더에는 상위 폴더가 없다고 가정하였으므로 시작 폴더를 반환한다.

```
>>> P1('../')
```

```
 '/'
```

예시 3) 마지막의 '/'를 제거하고 중간의 '/'는 '/'로 대체한다.

```
>>> P1('/home//foo/')
```

```
 '/home/foo'
```

예시 4) a 폴더 – 현재 폴더('.') - b 폴더로 갔다가 상위('..') – 상위('..') 폴더로 돌아오면 시작 폴더가 된다. 여기에서 다시 c 폴더로 들어가므로 '/c'가 된다.

```
>>> P1('/a/./b/../../c/')
```

```
'/c'
```

문제 2. 괄호 열고 닫기

'(', ')', '{', '}', '[', ']'으로 구성된 String 을 Input 으로 받고, 이 String 이 아래 규칙에 의해 유효한지를 판단하는 함수를 만들어라.

- 괄호가 열렸다면 같은 타입의 괄호에 의해 닫혀야 한다.
- 적절한 순서로 괄호가 배열되어야 한다.안쪽에서 열린 괄호가 먼저 닫히고, 바깥쪽 괄호가 닫혀야 한다.

예시 1) 유효한 입력이다.

```
>>> P2('()')
```

```
True
```

예시 2) 유효한 입력이다.

```
>>> P2('()[]{}')
```

```
True
```

예시 3) 안쪽에서 열린 '['가 닫히기 전에 ']'가 먼저 닫혔으므로 유효하지 않다.

```
>>> P2('([)]')
```

```
False
```

문제 3. Singly linked list 정렬하기

Singly linked list 의 Head node 가 주어졌을 때 이를 정렬하고 정렬된 Singly linked list 의 Head node 를 반환하여라. 여러가지 방법으로 구현할 수 있으나, Merge sort 의 방법론을 이용한 구현이 일부 뼈대 코드에 작성되어 있다. 뼈대 코드의 나머지 부분을 완성하여라.

- 현재 작성된 부분은 Merge sort 의 divide 부분이고, 과제로 작성해야 할 부분은 conquer 부분이다. (각 Division 을 sort 하고 Merge 하는 과정) Merge sort 의 방법론과 singly linked list 의 특성을 잘 이용하면 완성할 수 있다.
- Merge sort 방법론이 아닌 다른 방법론을 이용하고 싶다면 뼈대 코드를 수정해서 구현해도 괜찮으며, 결과가 맞게 나오면 정답으로 인정할 것이다. 단, Singly linked list 를 list 로 해체하여 정렬 후 다시 linked list 를 생성하는 방법 등은 인정하지 않는다.
- 함께 제공된 linked_list_helper.py 파일에 ListNode 의 정의와 구현 확인을 위해 활용할 수 있는 Helper function 들을 입력해 두었다.
 - create_linked_list: list 를 입력하면 list 의 순서대로 singly linked list 를 생성하고 head node 를 반환한다.
 - print_linked_list: singly linked list 의 head node 와 빈 list 를 입력하면 singly linked list 의 원소들을 리스트 형태로 출력해준다.
- 제출시 테스트 코드는 제거하고 제출한다.

예시 1)

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

```
>>> print_linked_list(P3(l1),[])
```

```
[1, 2, 3, 4]
```

예시 2)

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

```
>>> print_linked_list(P3(l2),[])
```

```
[-1, 0, 2, 3, 5]
```

예시 3)

```
>>> l3 = create_linked_list([1,3,3,1,3,1,3,3,2,3,2,2,1,1,1,3,2,2,1,1,2,2,2])
>>> print_linked_list(P3(l3),[])
[1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3]
```

문제 4. Linked list 순서 뒤집기

Singly linked list 를 입력받아 순서를 뒤집은 Singly linked list 로 만드는 함수를 작성하여라.
Linked list 의 각 Node 는 linked_list_helper.py 파일의 ListNode 정의를 활용하고, 함수의
return 값은 뒤집은 Linked list 의 첫번째 Node 로 한다.

- 제출시 테스트 코드는 제거하고 제출한다.

예시 1)

```
>>> l1 = create_linked_list([4,2,1,3])
>>> print_linked_list(P4(l1),[])
[3,1,2,4]
```

예시 2)

```
>>> l2 = create_linked_list([-1,5,0,2,3])
>>> print_linked_list(P4(l2),[])
[3,2,0,5,-1]
```

예시 3)

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

주의사항

- 코드를 Jupyter Notebook 에서 작성하였더라도 python 파일(.py)로 변환하여 제출할 것.
- 함수가 의도한 값을 Return 하는지를 확인. (Print 와 혼동하지 말 것)
- 파일명은 P1.py ~ P4.py 를 유지하고, 해당파일들을 PROJ1_학번_이름.zip 으로 압축하여 제출할 것. 예를 들면 학번이 2020-12345 이고, 이름이 Keondo Park 이라면

PROJ2_2020_12345_KeondoPark.zip 으로 압축하여 제출.

- 예시로 제시한 입력값 외에도 조교가 랜덤으로 생성한 입력값으로 코드가 잘 작성되었는지 테스트할 것이다.
- 채점은 프로그램에 의해 기계적으로 처리되므로 위 사항을 지키지 않은 경우 누락되거나 불이익을 받을 수 있음.
- 늦은 제출은 받지 않음.
- 표절검사를 수행하여 발각될 경우 성적 F 부여함.