

이산수학 2022-2 심화과제

목표

파이썬 환경에서 정렬, 탐색 알고리즘을 구현해 봅니다. 개념적으로 학습한 알고리즘을 코드로 구현하는 과정을 익히도록 합니다.

내용

- assignment.py 에 구현해야 하는 함수들이 정의되어 있습니다. 정렬, 탐색 알고리즘을 작성합니다.
- main.py 에는 테스트 코드가 주어져 있습니다. 알고리즘을 올바르게 구현하고 실행하면 다음과 같은 결과가 보일 것입니다.

```
1. Test Case -- Linear Search
PASSED
2. Test Case -- Binary Search
PASSED
3. Test Case -- Sort
PASSED
```

함수 명세

- `custom_linear_search(target_list, x)`
 - `target_list`에 대해 선형 탐색을 수행해 `x` 를 찾고 `x` 의 위치정보를 리턴하는 함수입니다.
 - (Input) `target_list`: 자료를 보관하고 있는 리스트 입니다.
 - (Input) `x`: 찾고자 하는 원소입니다.
 - (Output) `x_index`: `target_list` 내 `x`의 위치정보를 리턴합니다. 만약 `x` 가 `target_list`에 없다면 -1을 리턴합니다.
 - (Output) `num_steps`: 탐색에 사용된 반복 loop 횟수를 반환합니다.
- `custom_binary_search(target_list, x)`
 - `target_list`에 대해 이진 탐색을 수행해 `x` 를 찾고 `x` 의 위치정보를 리턴하는 함수입니다.
 - (Input) `target_list`: 자료를 보관하고 있는 리스트 입니다.
 - (Input) `x`: 찾고자 하는 원소입니다.
 - (Output) `x_index`: `target_list` 내 `x`의 위치정보를 반환합니다. 만약 `x` 가 `target_list`에 없다면 -1을 리턴합니다.
 - (Output) `num_steps`: 탐색에 사용된 반복 loop 횟수를 반환합니다.
- `custom_sort(target_list)`
 - `target_list`를 정렬해 반환하는 함수입니다.
 - (Input) `target_list`: 자료를 보관하고 있는 리스트 입니다.
 - (Output) `target_list`: 정렬된 list를 반환합니다.
 - (Output) `num_steps`: 정렬에 사용된 loop의 반복횟수를 반환합니다.

주의 사항

- 코드 구현 시, 주석으로 ("YOU ARE ONLY ALLOWED TO MODIFY THIS ZONE") 이라고 되어 있는 부분에만 코드를 작성할 수 있습니다. 따라서, 함수의 입/출력을 임의로 수정할 수 없고 main.py 를 수정할 수 없습니다.
- main.py는 예시로 주어진 테스트 코드입니다. 실제로는 다른 종류의 데이터를 다루는 코드가 테스트를 위해 사용될 것입니다. 임의의 데이터 입력에 대해 동작하도록 코드를 작성해야 합니다.
- 함수의 입력으로 사용될 리스트는 정수, 실수, 문자, 문자열 등 비교 가능한 임의의 데이터를 보관할 수 있습니다. 길이가 달라질 수 있습니다.
- sorted(), index(), 등 정렬, 탐색 결과를 제공하는 함수를 사용할 수 없습니다. 반드시, 반복문, 조건문, 비교연산, 할당연산 등 기본적인 연산을 이용해 함수를 구현해야 합니다.
- 반드시 강의에서 배운 정렬 알고리즘 중 하나를 사용해야 합니다. 그렇지 않으면 정렬 케이스는 통과하지 못한 것으로 간주합니다.
- 테스트 케이스 통과 여부에 따라 환산된 점수가 과제 점수로 부여될 것입니다 (100점 만점, 성적반영비율 20%). 임의로 코드를 수정하거나 정렬, 탐색 결과를 자동으로 제공하는 함수를 사용할 경우 과제 점수는 0점입니다.

참고: num_steps 지정 방법 예시

while 문 사용 시

```
num_steps=0
while i < j:
    # some codes
    num_steps += 1

return num_steps
```

for 문 사용 시

```
num_steps=0
for i in range(10):
    for j in range(10):
        # some codes
        num_steps += 1

return num_steps
```