

과제

장유선

2023.08.22

1. 문제 정의

Single Linked List를 구현하고, python에 내장된 list와 0번째 Index에 데이터를 1천만번 Insert하는 시간을 비교하세요

Python 내장 List에서 0번째 Index에 1천만번 Insert

```
l = []
for i in range(10000000):
    l.insert(0, i)
```

=> 시간 측정이 오래걸려 10만번으로 수정

2. Python Code Hard Copy

```
import time

# 노드 클래스 정의
class Node:
    def __init__(self, data):
        self.data = data
        self.next = None

# 연결 리스트 클래스 정의
class Single_LinkedList:
    def __init__(self):
        self.front = None
        self.rear = None

    def insert(self, data):
        new_node = Node(data)
        if self.front is None:
            self.front = self.rear = new_node
        else:
            new_node.next = self.front
            self.front = new_node

# Python 내장 리스트와 연결 리스트 초기화
py_list = []
linked_list = Single_LinkedList()
```

```

# Python 내장 리스트
start_time = time.time()
for i in range(100000):
    py_list.insert(0, i)
end_time = time.time()
py_list_time = end_time - start_time

# Single Linked List
start_time = time.time()
for i in range(100000):
    linked_list.insert(i)
end_time = time.time()
linked_list_time = end_time - start_time

print(f"Python 내장 리스트에 데이터를 삽입하는 데 걸린 시간: {py_list_time} 초")
print(f"Single Linked List 에 데이터를 삽입하는 데 걸린 시간: {linked_list_time} 초")

```

3. Code 설명

3-1. `Node` 클래스 정의

```

class Node:
    def __init__(self, data):
        self.data = data
        self.next = None

```

- `Node` 클래스는 연결 리스트의 노드를 나타낸다.

- `__init__` 메서드는 노드를 초기화하고 데이터를 저장하며, `next` 포인터는 다음 노드를 가리킴.

3-2. `Single_LinkedList` 클래스 정의

```

class Single_LinkedList:
    def __init__(self):
        self.front = None
        self.rear = None

    def insert(self, data):
        new_node = Node(data)
        if self.front is None:
            self.front = self.rear = new_node
        else:
            new_node.next = self.front
            self.front = new_node

```

- `Single_LinkedList` 클래스는 Single Linked List를 나타낸다.
- `__init__` 메서드는 연결 리스트를 초기화하고, `front`와 `rear` 포인터를 `None`으로 설정
- `insert` 메서드는 주어진 데이터를 연결 리스트의 맨 앞에 삽입

3-3. 데이터 삽입

```
py_list = []
linked_list = Single_LinkedList()

start_time = time.time()
for i in range(100000):
    py_list.insert(0, i)
end_time = time.time()
py_list_time = end_time - start_time

start_time = time.time()
for i in range(100000):
    linked_list.insert(i)
end_time = time.time()
linked_list_time = end_time - start_time
```

- `for` 루프를 사용하여 0부터 99,999까지의 숫자를 삽입
- 시간 측정을 위해 루프 전후에 `time.time()`을 사용하여 시작 및 종료 시간을 기록 및 소요 시간 계산

4. 결과

Python 내장 리스트에 데이터를 삽입하는 데 걸린 시간: 2.3702259063720703 초
 Single Linked List에 데이터를 삽입하는 데 걸린 시간: 0.2537405490875244 초

5. 결과 화면

```
Python 내장 리스트에 데이터를 삽입하는 데 걸린 시간: 2.3702259063720703 초
Single Linked List에 데이터를 삽입하는 데 걸린 시간: 0.2537405490875244 초
```