

1. 제목

Chat GPT를 활용한 자료구조 선택과 최적화

2. 이름

김태훈 (Theodore Kim)

3. 제출일

24-10-14

4. 과제 목표

스스로 생각하는 과제의 목표를 작성해 주세요.

자료구조의 선택과 최적화

효율적인 자료구조를 선택하고 이를 통해 대용량 데이터를 빠르게 처리하는 방법을 학습하는 것이 목표입니다. ChatGPT를 활용하여 파이썬 코드 작성에 도움을 받고, 선택한 자료구조의 장단점을 학습합니다.

5. ChatGPT 활용

👉 질문 내용

"대량의 데이터를 빠르게 검색하고 업데이트할 수 있는 자료구조는 무엇이 좋을까요?"
"충돌을 최소화할 수 있는 해시 테이블을 어떻게 구현할 수 있나요?"

👉 ChatGPT 답변 요약

ChatGPT는 해시 테이블이 평균적으로 상수 시간 복잡도($O(1)$)에서 검색과 업데이트가 가능해 대용량 데이터를 처리하는 데 매우 적합하다고 설명했습니다. 충돌을 해결하는 방법으로는 체이닝(chaining)을 통해 각 해시 슬롯에 연결 리스트를 두어 충돌을 해결하는 방식을 권장했습니다.

6. 자료구조 선택 이유

해시 테이블을 선택한 이유는 빠른 검색 및 업데이트가 가능하다는 점입니다. 특히 대량의 데이터에서 효율적으로 동작하기 때문에 적합하다고 판단했습니다. 체이닝을 사용한 해시 충돌 해결 방식을 통해 데이터 처리의 안정성을 높일 수 있었습니다.

7. 코드 작성 과정

1. 해시 함수를 정의하고 데이터를 해시 테이블에 저장할 수 있도록 기본 구조를 설계.
2. 충돌 해결을 위한 체이닝 기법을 구현.
3. 데이터 검색 및 업데이트 기능을 추가하여 해시 테이블의 기능을 완성.

해시 테이블 코드 예시:

```
class HashTable:
    def __init__(self):
        self.size = 10
        self.table = [[] for _ in range(self.size)]

    def _hash(self, key):
        return hash(key) % self.size

    def insert(self, key, value):
        index = self._hash(key)
        for kvp in self.table[index]:
            if kvp[0] == key:
                kvp[1] = value
                return
        self.table[index].append([key, value])

    def search(self, key):
        index = self._hash(key)
        for kvp in self.table[index]:
            if kvp[0] == key:
                return kvp[1]
        return None

    def delete(self, key):
        index = self._hash(key)
        for i, kvp in enumerate(self.table[index]):
            if kvp[0] == key:
                del self.table[index][i]
                return True
        return False

# 해시 테이블 생성 및 데이터 삽입
ht = HashTable()
ht.insert("name", "Anne")
ht.insert("age", 16)

# 데이터 검색
print(ht.search("name")) # Anne 출력
print(ht.search("age"))  # 16 출력
```

```
# 데이터 삭제
ht.delete("age")
print(ht.search("age")) # None 출력
```

8. 코드 실행 결과

1. `ht.search("name")` 결과: **Anne**
2. `ht.search("age")` 결과: **16**
3. `ht.delete("age")` 후 `ht.search("age")` 결과: **None**

실행 결과는 예상대로 작동했으며, 해시 충돌 시 연결 리스트로 해결됨을 확인했습니다.

9. 문제 해결 과정 및 배운 점

ChatGPT와 상호작용하며 해시 테이블의 개념을 깊이 이해할 수 있었습니다. 특히, 해시 충돌을 해결하는 체이닝 기법을 적용하여 실전에서의 해시 테이블 최적화 방식을 배울 수 있었습니다. 또한, 해시 테이블의 $O(1)$ 검색 속도의 이점을 통해 대용량 데이터를 효율적으로 처리할 수 있음을 알게 되었습니다. 앞으로 다양한 데이터 처리 상황에 맞춰 다른 자료구조도 고려해 볼 계획입니다.