

밀라 알고리즘 스터디 Week 01

1주차 과제 hyundo-jung

스택, 큐, 리스트, 해시맵에 대한 개념, 주요연산, 시간복잡도 정리

스택 (stack): 항목들이 쌓여있는 구조, LIFO (Last-in-first-out) → 가장 마지막에 추가된 항목이 가장 먼저 제거된다.

- 주요연산 (operation):
 - push: 스택 맨 위에 새로운 항목 추가
 - 시간복잡도: $O(1)$
 - pop: 스택 맨 위에 있는 항목 반환
 - 시간복잡도: $O(1)$
 - isEmpty: 스택이 비었는지 확인
 - 시간복잡도: $O(1)$
 - search: 특정한 값이 존재하는지
 - 시간복잡도: $O(n)$
- 구현 (implementation)
 - vector with capacity, linked list
- 탐색 (search)

큐 (queue): 스택과 비슷하지만 FIFO (First-in-first-out) → 제일 처음에 들어온 데이터가 먼저 나간다.

- 주요연산:
 - enqueue: 큐 마지막에 데이터 추가
 - 시간복잡도: $O(1)$
 - dequeue: 큐 시작에서 데이터 반환
 - 시간복잡도: $O(1)$
 - isEmpty: 큐가 비었는지 확인
 - 시간복잡도: $O(1)$
 - search: 특정한 값이 존재하는지
 - 시간복잡도: $O(n)$
- 구현 (implementation)
 - linked list, ring buffer

리스트 (list): 데이터가 일렬로 나열되어있고 순서가 있다

- 주요연산
 - len: 전체 요소의 개수 반환
 - 시간복잡도: $O(1)$
 - count(elem): elem의 개수 반환
 - 시간복잡도: $O(n)$
 - index(elem): elem의 인덱스 반환
 - 시간복잡도: $O(n)$
 - a.sort(): 리스트 정렬
 - 시간복잡도: 활용한 정렬에 따라 다르다. 주로 $O(n \log n)$
- 순차 리스트 (Array list): 배열을 기반으로 한 리스트
 - 인덱스를 기반으로 접근할 수 있다 → 속도가 빠르다
 - 배열 기반 → 리스트 크기가 제한된다 + 중간에 데이터 삽입시 기존 데이터들 재비치
- 연결 리스트 (linked list): 포인터를 사용하여 리스트
 - 한 노드는 데이터 그리고 다른 노드를 향한 포인터로 이루어져있다
 - 크기가 제한되지 않고 중간에 데이터를 쉽게 삽입/삭제할 수 있다
 - 구현이 복잡하고 탐색할때 순차 리스트보다 비효율적이다
 - 연결 리스트의 종류들: 단일 연결 리스트(singly linked list), 원형 연결 리스트(circular linked list), 이중 연결 리스트(doubly linked list)



해시맵 (Hash Map): 키(key)와 값(value)쌍으로 이루어진 자료 구조

- 해싱함수: 키는 받아서 해시코드(hashcode)를 반환
 - 해시코드는 값을 저장하는 배열의 인덱스값
 - 주로 인덱스값은 해시코드 % 저장용량
- 특징들:

- 키는 중복될 수 없다
- null 키와 null 값을 저장할 수 있다.
 - 키는 중복될 수 없기에 하나만 가능
- 주요연산:
 - 해시맵의 해싱 특성상 평균 시간복잡도와 최악의 경우 시간복잡도로 나뉜다. 최악의 경우는 해싱 충돌이 자주 일어날때를 얘기한다.
 - put: 삽입
 - 시간복잡도: $O(1)$ & $O(n)$
 - get: 검색 및 반환
 - 시간복잡도: $O(1)$ & $O(n)$
 - remove: 키-값 쌍 삭제
 - 시간복잡도: $O(1)$ & $O(n)$
 - mem?: 키 존재 여부
 - 시간복잡도: $O(1)$ & $O(n)$
 - size: 키-값 쌍 개수 반환
 - 시간복잡도: $O(1)$
 - 주로 length, size라는 변수 존재
- 해싱 충돌 (hash collision): 다른 키가 같은 해시코드를 반환
 - 해결방법:
 - 체이닝 (chaining): 버킷을 연결 리스트로 구현하여 한 버킷이 여러 값을 갖는다
 - 개방 주소법 (Open Addressing): 충돌이 발생하면 다른 버킷의 위치에 삽입 시도
→ linear probing, quadratic probing, double hashing
 - 버킷 확장 (Bucket Expansion): 버킷 크기를 확정해 여러 키-값 쌍을 저장할 수 있게 한다.