**2과목 소프트웨어 개발**

1장 데이터 입•출력 구현

036 자료구조

- 자료를 기억장치의 공간 내에 저장하는 방법, 저장된 그룹 내에 존재하는 자료 간의 간계, 처리 방법을 연구 분석하는 것

선형 구조 : 배열 / 선형리스트(연속, 연결 리스트) / 스택 / 큐 / 데크

비선형 구조 : 트리 / 그래프

배열 – 동일한 자료형의 데이터들이 같은 크기로 나열되어 순서를 가지고 있는 집합

선형리스트 – 일정한 순서에 의해 나열된 자료구조

-> 연속리스트 – 배열과 같이 연속되는 기억장소에 저장되는 자료구조

-> 연결리스트 – 순서에 따라 노드의 포인터 부분을 이용하여 서로 연결시킨 자료구조

스택 – 리스트의 한쪽 끝으로만 자료의 삽입, 삭제 작업이 이루어지는 자료 구조 / 후입선출

큐 - 리스트의 한쪽 끝으로만 자료의 삽입 작업이 이루어지고 다른 한쪽에서 삭제 작업이 이루어짐 / 선입선출

037 트리

정점과 선분을 이용하여 사이클을 이루지 않도록 구성한 그래프의 특수한 형태

노드 / 근 노드 / 디그리 / 단말노드 / 자식 노드 / 부모 노드 / 형제 노드 / 트리의 디그리

트리의 운행법

Preorder Inorder Postorder

Root -> Left -> Right Left -> Root -> Right Left -> Right -> Root

수식의 표기법

PreFix InFix PsotFix

연산자 -> Left -> Right Left -> 연산자 -> Right Left -> Right -> 연산자

038 정렬

삽입 정렬 – 이미 순서화된 파일에 새로운 하나의 레코드를 순서에 맞게 삽입시켜 정렬

평균과 최악 모두 수행 시간 복잡도 O()

이미 순서화된 파일에, n번째 키를 앞의 n-1개의 키와 비교

쉘 정렬 – 매개 변수의 값으로 서브파일을 구성하고 서브파일을 삽입 정렬 방식으로 순서 배열하는 과정을 반복하는 정렬 방식

평균 수행 시간 복잡도 O() 최악의 수행 시간 복잡도 O()

매개변수

선택 정렬 - n개의 레코드 중에 최소값을 찾아 첫 번째 레코드 위치에 놓고 나머지 중에서 다시 최솟값을 찾아 두 번째 레코드 위치에 놓는 방식을 반복하여 정렬하는 방식

평균과 최악 모두 수행 시간 복잡도 O()

n개의 레코드 중에서 최소값을 찾아서

버블 정렬 – 인접한 두 개의 레코드 키 값을 비교하여 그 크기에 따라 레코드 위치를 서로 교환하는 정렬 방식

평균과 최악 모두 수행 시간 복잡도 O()

인접한 두 개의 레코드

퀵 정렬 – 하나의 파일을 부분적으로 나누어 가면서 정렬하는 방식

평균 수행 시간 복잡도 O) 최악의 수행 시간 복잡도 O()

하나의 파일을 부분적으로 나누어

힙 정렬 – 전이진 트리를 이용한 정렬 방식

평균과 최악 모두 수행 시간 복잡도 O)

전이진 트리

2-Way 합병 정렬 - 이미 정렬되어 있는 두 개의 파일을 한 개의 파일로 합병하는 정렬방식

평균과 최악 모두 수행 시간 복잡도 O)

이미 정렬된 두 개의 파일을 한 개의 파일로

기수 정렬 – 큐를 이용하여 자릿수별로 정렬하는 방식

평균과 최악 모두 수행 시간 복잡도 O(dn)

버킷

039 검색 – 이분 검색 / 해싱

이분 검색 – 전체 파일을 두 개의 서브파일로 분리해가며 key 레코드를 검색하는 방식

찾고자 하는 key 값을 파일의 중간 레코드 key 값과 비교하면서 검색

중간 레코드 번호 M =

해싱 – 해시 테이블이라는 기억공간을 할당하고 해시함수를 이용하여 레코드 키에 대한 해시 테이블 내의 홈 주소를 계산한 후 주어진 레코드를 해당 기억장소에 저장하거나 검색 작업을 수행하는 방식

해시 테이블 – 버킷 / 슬롯 / collision / Synonym / Overflow

해싱함수 – 제산법 / 제곱법 / 폴딩법 / 기수 변환법 / 대수적 코딩법 / 숫자 분석법 / 무작위법

040 데이터베이스 개요

데이터 저장소 – sw 개발 과정에서 다루어야 할 데이터들을 논리적인 구조로 조직화하거나 물리적인 공간에 구축한 것을 의미

데이터베이스 – 통합된 데이터 / 저장된 데이터 / 운영 데이터 / 공용 데이터

DBMS – 사용자와 데이터베이스 사이에서 사용자의 요구에 따라 정보를 생성해주고 관리해주는 소프트웨어

기능 – 정의 / 조작 / 제어

스키마 – 데이터베이스의 구조와 제약 조건에 관한 전반적인 명세를 기술한 메타데이터의 집합

사용자 관점 - 외부 스키마 / 개념 스키마 / 내부 스키마

041 데이터 입•출력

데이터 입•출력 – 소프트웨어 기능 구현을 위해 데이터베이스에 데이터를 입력하거나 데이터베이스의 데이터를 출력하는 작업을 의미

sql을 사용

데이터 접속 – 개발 코드 내에 SQL 코드를 삽입하거나 객체와 데이터를 연결하는 것

SQL – 국제표준 데이터베이스 언어

데이터 정의어(DDL) / 데이터 조작어(DML) / 데이터 제어어(DCL)

데이터 접속 – 소프트웨어의 기능 구현을 위해 프로그래밍 코드와 데이터베이스의 데이터를 연결하는 것

SQL Mapping / ORM

트랜잭션 – 데이터베이스의 상태를 변환시키는 논리적 기능을 수행하기 위한 작업의 단위

COMMIT / ROLBACK / SAVEPOINT

042 절차형 SQL

프로그래밍 언어와 같이 연속적인 실행이나 분기, 반복 등의 제어가 가능한 SQL

데이터베이스 전용의 간단한 프로그래밍이라고 할 수 있음

프로시저 – 특정기능을 수행하는 일종의 트랜잭션 언어, 호출을 통해 실행되며 저장해 놓은 SQL 작업 수행

트리거 – 데이터베이스 시스템에서 데이터 입력, 갱신, 삭제 등의 이벤트가 발생할 때마다 관련 작업이 자동으로 수행

사용자 정의 함수 – 프로시저와 유사하게 SQL을 사용하여 일련의 작업을 연속적으로 처리, 종료시 결과를 단일값으로 반환

디버깅을 통해 기능의 적합성 여부를 검증하고 실행을 통해 결과를 확인하는 테스트 과정 수행

쿼리 성능 최적화 – 데이터 입•출력 애플리케이션의 성능 향상을 위해 SQL 코드를 최적화하는 것