1. Big-O 표기법

* 수행되는 연산(산술, 비교, 대입)등의 개수를 대략적으로 판단\
* Big-O 표기법 1단계 : 대략적인 계산

텍스트이(가) 표시된 사진

자동 생성된 설명

* Big-O 표기법 2단계 : 대장만 남긴다!
* 영향력이 가장 큰 대표 항목만 남기고 삭제 (4n^2)
* 무시 (4무시)

텍스트이(가) 표시된 사진

자동 생성된 설명

* 표기법 의의

테이블이(가) 표시된 사진

자동 생성된 설명

1. 선형 vs 비선형

* 선형 구조는 자료를 순차적으로 나열한 형태 (배열, 연결 리스트, 스택, 큐)

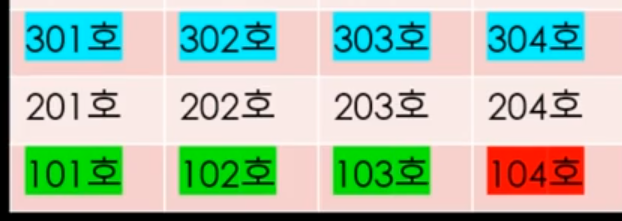
Ex) 은행에서 줄을 서서 기다리는 것과 같이 순차적임

* 비선형 구조는 하나의 자료 뒤에 다수의 자료가 올 수 있는 형태(트리, 그래프)

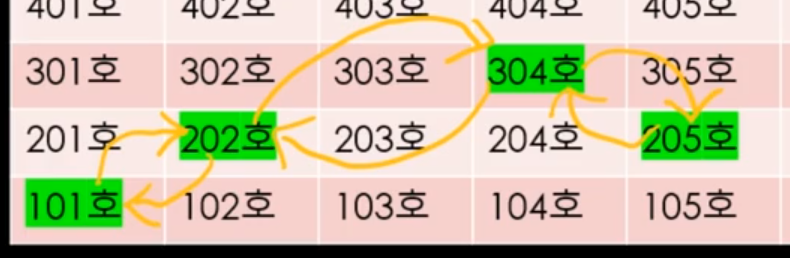
Ex) 폴더에 들어가면 여러 폴더가 있는 것과 같음

1. 배열 vs 동적 배열 vs 연결 리스트 (호텔로 비교

* 배열 : 사용할 방 개수를 고정해서 계약하고, 연속되 방으로 배정 받아 사용
* 장점 : 연속적임 // 단점 : 방을 추가하거나 축소불가, 유동적으로 사용할 수 없음
* 동적 배열 : 사용할 방 개수를 유동적으로 계약, 연속된 방으로 배정 받아 사용
* 장점 : 연속적임, 배열 보다 유동적 // 단점 : 연속적이기 때문에 아래와 같이 101-102-103 호가 배정된 상황에서 104호를 추가 배정하고 싶지만, 이미 선정이 되어 있다면 다른 위치에서 연속적으로 사용해야함 => 이사 비용에 대한 문제(최적화), 하지만 연속적이기 때문에 중간의 삽입, 삭제가 어려움
  + 동적 배열 할당 적책 : 실제로 사용할 방보다 많이, 여유분을 두고 예약(1.5~2배)



* 연결 리스트 : 연속되지 않은 방을 사용
* 장점 : 중간 추가/삭제 이점 // 단점 : n번째 방을 바로 찾을 수가 없음(갑자기 3번째 방이 누구냐? 물어보면 넘겨넘겨 찾아야함)



* 맵을 만들 때, 사용할 데이터 선택하기

텍스트이(가) 표시된 사진

자동 생성된 설명

1. 동적 배열 구현하기

* List<> : 삽입, 호출, 삭제 문법



텍스트이(가) 표시된 사진

자동 생성된 설명

* 코드로 리스트 만들기

텍스트이(가) 표시된 사진

자동 생성된 설명

1. 연결 리스트 구현하기

* LinkedList<>

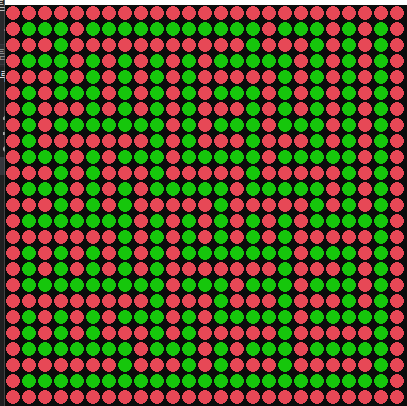
텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

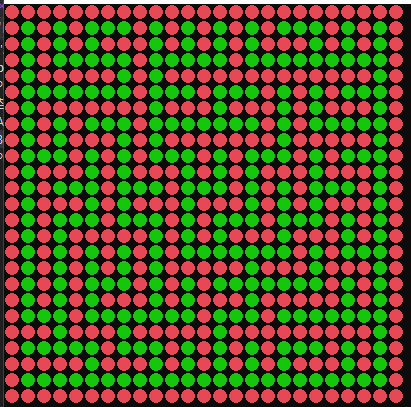
자동 생성된 설명

1. 길찾기 알고리즘

* BinaryTree



* SideWinder



1. 우수법 길찾기 알고리즘