

2장 파이썬에 대해서 묻는 것 X

설명, 빈칸채우기(ex, 시간복잡도)

1장

빈칸(시간복잡도, 조직화 이유)

선형, 비선형(그래프, 트리)

알고리즘

자연어-> 애매

흐름도-> 시간오래걸림

유사코드-> 바로 안돌아감

파이썬 표현 -> 바로 잘 돌아감

일반함수 문제점 => 예를 들어 insert 구현(여러 insert 함수가 많이 나옴 -> 낭비) (함수지향적)

실행시간 -> 타임모듈 time.time

시간복잡도 개념

파이썬 리스트도 클래스임

차수가 클수록 n 이 커질수록 대부분의 영향을 미침(점근적 표기법, 빅오 표기법)

빅오 표기법 -> 개념만, (증명은 필요X)

시간복잡도 문제 출제(ex, 답) $O(n)$

빅세타 표기법(상한이면서 하한)

빅오메가(하한)

빅세타로 생각하고 문제 풀어야 함

최선 최악의 평균 $O(n)$ -> 인터넷에 찾아보기

순환 -> 반복, 반복 -> 순환

팩토리얼(순환, 반복 모두 가능)

거듭제곱(순환) 무조건 나옴

시간 복잡도 $O(n)$, $O(\log n)$ (훨씬 좋은 함수), $O(n^2)$ -> 뭐가 더 빠른지 비교하는 문제 무조건 나

옴

동적 계획법? -> 피보나치 밑에서부터 쌓는 것?

피보나치는 반복이 훨씬 빠름

하노이탑 순환(시험에 X) -> $O(2^n)$ 만 알아두기

2장

파이썬은 모든 변수가 포인터임 (132라는 객체를 number라는 포인터가 가리킴)

2장은 시험X?

3장

파이썬은 따로 배열이 X

배열 대신 리스트와 튜플

집합(중복x, 순서x) -> 리스트와의 차이점

인접 행렬 -> 배열 구조

인접 리스트 -> 연결된 구조

리스트 구현 방법(빈칸 채우기 많이 나옴)

파이썬 리스트의 시간 복잡도

라인편집기 시험문제 X

4장

스택, 큐 시험 많이 출제!!!!

~~의 용도 -> 시험 출제

스택의 용도!!

연습문제

공백 리스트를 스택으로 씀(책 예제)

스택의 구현(함수 버전, 클래스 버전) -> 시험문제

스택은 구현보다는 활용이 중요

스택의 응용 문제 출제

스택 괄호검사!!!!(식만 바뀌서 출제)

소스파일 괄호검사 통과 ~

계산기 프로그램 출제

중위 표기 -> 후위 표기 변환 -> 후위 계산**

후위표기 수식의 계산 방법(스택에 숫자가 들어가야 함)

중위-> 후위 변환(스택에는 연산자와 열리는 괄호만 들어감)

후위 표기는 괄호가 없음(이미 우선순위가 반영이 되어있음) -> 중위표기와의 차이점

중위, 후위 -> 코드 내기는 어려움, 그림 스택은 출제, 식만 바꾸어 출제

미로 탐색 출제

탐색 순서(큐에서 설명)

깊이 우선 -> 스택

너비 우선 -> 큐

이웃 화소 검사 순서에 따라 큐에 들어가는 순서가 다름(답이 달라짐) -> 시험 문제 출제

스택은 개념, 그림 그리는 부분 많이 출제

5장

큐(선입선출)의 응용 잘 보기

인생이 큐..ㅋ//ㅋ..ㅋ.ㅋ.

우리가 사용하는 것의 90% 큐

선형 큐의 문제점 -> 원형 큐가 해결

원형 큐 -> 공백, 포화 시험 문제 단골

원형 큐 구현

5장 18p 깊이우선 너비우선 미로 탐색 모두 출제

덱 -> 원형 큐에서 추가된 연산

덱 구현 -> 읽어보기

우선순위 큐 -> 굳이 정렬되어 있지 않아도 됨

힙 -> 꺼낼 때 루트를 없앴

우선순위 큐 -> 정렬 O, 정렬 X (시간 복잡도)

전략적 미로탐색 해봤죠..?

6장

배열 구조와 연결된 구조의 차이, 장단점, 특징 -> 필수로 시험 출제

연결 리스트-> 노드가 꼭 필요함

연결 리스트 종류

연결된 스택 연산들 -> 삽입, 삭제, 전체 노드의 방문(코드 봐두기)

단순연결리스트 응용 연결 리스트(삽입/삭제하고 싶은 위치의 이전 노드를 알아야 함, 그림, 코드)

원형연결리스트 -> front/tail 둘 중 무엇을 사용하는 것이 더 유리? (설명**, 그림, rear == tail, front == tail.link)

링크 그림 -> 그렇게 많이는 안 나옴(하나 정도는 나옴)

이중연결리스트 응용: 연결된 덱 -> 후단 삭제 힘들 ->

이중 연결 리스트 무조건 시험 출제

7장

레코드, 선택정렬, 삽입정렬 -> 특징

선택정렬 - 알고리즘이 간단, 자료 이동 횟수가 미리 결정됨 $O(n^2)$

삽입정렬 - 많은 이동 필요 -> 레코드가 큰 경우 불리, 안정된 정렬방법, 대부분 정렬되어 있으면 매우 효율적(최선: $O(n)$, 최악: $O(n^2)$, 평균: $O(n^2)$)

버블정렬- 최선 최악 평균 모두 $O(n^2)$, 레코드의 이동 과다(최악: 이동횟수 = $3 \times$ 비교횟수, 최선: 이동횟수 = 0, 평균 $O(n^2)$)

정렬응용: 집합 (3장은 리스트를 이용해 집합 표현(정렬X))

정렬하여 집합했을 때 어떤 장점, 효율이 있는지 -> 문제 출제(비교, 합집합, 교집합, 차집합 n^2 -> n)

탐색과 맵 구조 개념

순차 탐색(정렬x, 평균 비교횟수 $(n+1)/2$, 최악 비교횟수 n , 시간 복잡도 $O(n)$)

이진탐색(정렬o, 시간 복잡도 $O(\log_2 n)$, 순환, 반복 구현 가능)

보간탐색($low + (high-low) * (key - a[low].key) / (a[high].key - a[low].key)$)

해싱(키 값에 대한 산술적 연산에 의해 테이블의 주소를 계산)

해시 테이블 개념(키 값의 연산에 의해 직접 접근이 가능한 구조)

선형 조사법이 뭔지 나옴(설명): 충돌이 일어나면 해시 테이블의 다음 위치에 저장 + 군집화 현상

선형 조사법에서 삭제할 때 어떤 문제가 발생하는지 (설명)

- ➔ 삭제 후 기존 빈 버킷과 똑같이 놔두면 다른 키 탐색 시 해당 부분에서 멈추므로 오류 발생, 따라서 빈 버킷을 두 가지로 분류하여 삽입되었다가 삭제된 곳과 아예 비어있는 곳 두가지로 나누어야 함

군집화 완화 방법: 이차 조사법(제공하여 더한만큼 이동), 이중 해싱법(재해싱, 충돌 시 다른 해시 함수 사용)

체이닝 개념 확인(코드x) -> 하나의 버킷에 여러 개의 레코드를 저장할 수 있도록 하는 방법

해시 함수(어떤 해시함수가 좋은 지 보기)

- ➔ 충돌이 적어야 한다.
- ➔ 함수 값이 테이블의 주소 영역 내에서 고르게 분포되어야 한다. (군집화x)
- ➔ 계산이 빨라야 한다.

나의 단어장x

(중간 제공 함수, 비트 추출 함수, 숫자 분석 함수, 탐색키가 문자열인 경우)

순차탐색 - $n, 1, n$

이진탐색 - $\log n, n, n$

이진 균형 탐색 트리 - $\log n, \log n, \log n$

이진 경사 탐색 트리 - n, n, n

해싱 최선 - $1, 1, 1$

해싱 최악 - n, n, n

8장

트리 그림 하나 그려서 트리 용어 빈칸 채우기(트리 개념)

일반 트리 표현 방법(다른 방법) -> 시험 출제

일반 트리는 차수가 2보다 큰 것

이진 트리(완전 이진 트리, 포화 이진 트리 당연히 알아야 함)

이진 트리의 성질 알아야 함

이진 트리 (배열, 연결된 구조)

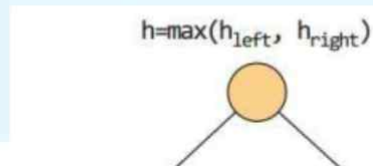
순회(전위 중위 후위 레벨) 순서 그려라 무조건 시험 출제

이진 트리 높이(코드 어떻게 구현하는지 알기)

• 트리의 높이

```
def calc_height(n) :  
    if n is None :  
        return 0  
    hLeft = calc_height(n.left)  
    hRight = calc_height(n.right)  
    if (hLeft > hRight) :  
        return hLeft + 1  
    else:  
        return hRight + 1
```

공백 트리 --> 0을 반환
왼쪽 트리의 높이 --> hLeft
오른쪽 트리의 높이 --> hRight
더 높은 높이에 1을 더해 반환.



모스코드(왜 결정 트리를 만드는지 하는지 설명하기)

➔ 시간 복잡도를 줄이기 위해(순차탐색으로 할 경우 $O(n)$, 결정트리 $O(\log n)$)

힙 시험에 나옴 (ex, 힙, 이진탐색트리 숫자 6-7개 -> 힙이 어떻게 만들어지는지(그림그리기))

힙 -> 루트를 삭제하면 어떻게 바뀌는지

이진 탐색 트리 -> 리스트 주어진다면 이때 만들어지는 이진 탐색 트리를 그려라(중위 순회 방식으로 적어라)

Upheap($O(\log n)$), Downheap($O(\log n)$) (큰 자식이랑 비교)

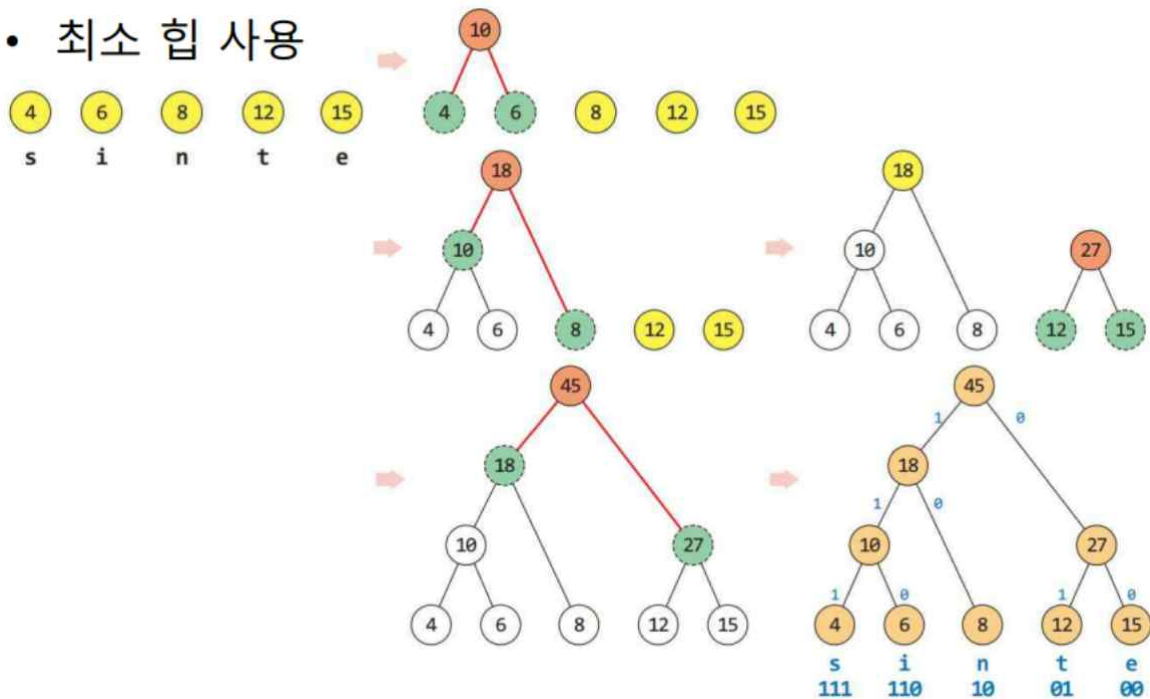
Meanheap(작은 자식)

힙을 왜 배열구조로 나타내는 것이 좋은지(완전 이진 트리, 각 노드에 번호 붙임, 인덱스 0 사용)

힙의 복잡도 분석(삽입($O(\log n)$), 삭제($O(\log n)$) 연산)

허프만 코드(빈도를 주어 주고 그려보라는 문제 출제)

• 최소 힙 사용



9장

AVL 트리 빼기 but, AVL 트리가 무엇인지는 나옴, 균형이 잡힌 트리를 위해 AVL트리가 나옴

뭐가 주어지면 탐색트리가 만들어지는 과정

탐색트리 삽입, 삭제, 탐색 연산 3가지 경우(코드X, 개념으로 공부하기)

이진 탐색 트리에서는 코드는 안나옴!!

10장

그래프 문제 개념, 특징

인접, 정점, 집합

그래프 주어지고 깊이우선탐색, 너비우선탐색 순서를 적어라

B, C, D가 주어지면 알파벳 순으로 B부터 먼저 선택해야 함(시험 문제 풀 때 약속!!)

연결 성분, 신장 트리, 위상 정렬

위상 정렬(하나에 대해서 여러 가지의 결과가 나올 수 있음)

11장

가중치 그래프

최소 비용 신장 트리(그래프 주어지고 각 알고리즘에서 뭐가 찾아지는 과정 적을 수 있어야 함)

플로이드에서 하나씩 나타내기는 어려움

다익스트라에서는 거리가 확정된 정점 표로 만들어서 분반마다 가중치만 좀 바뀌 문제 출제

크러스칼, 프림 -> 진행되는 과정

수업시간에 했던 내용 출제

서비스 센터의 콜 큐

프린터와 컴퓨터 사이의 인쇄 작업 큐

실시간 비디오 스트리밍

시뮬레이션의 대기열(공항, 은행)

통신에서의 데이터 패킷 모델링

정렬($o1, on, on, on$)

정렬($on, o1, o1$)