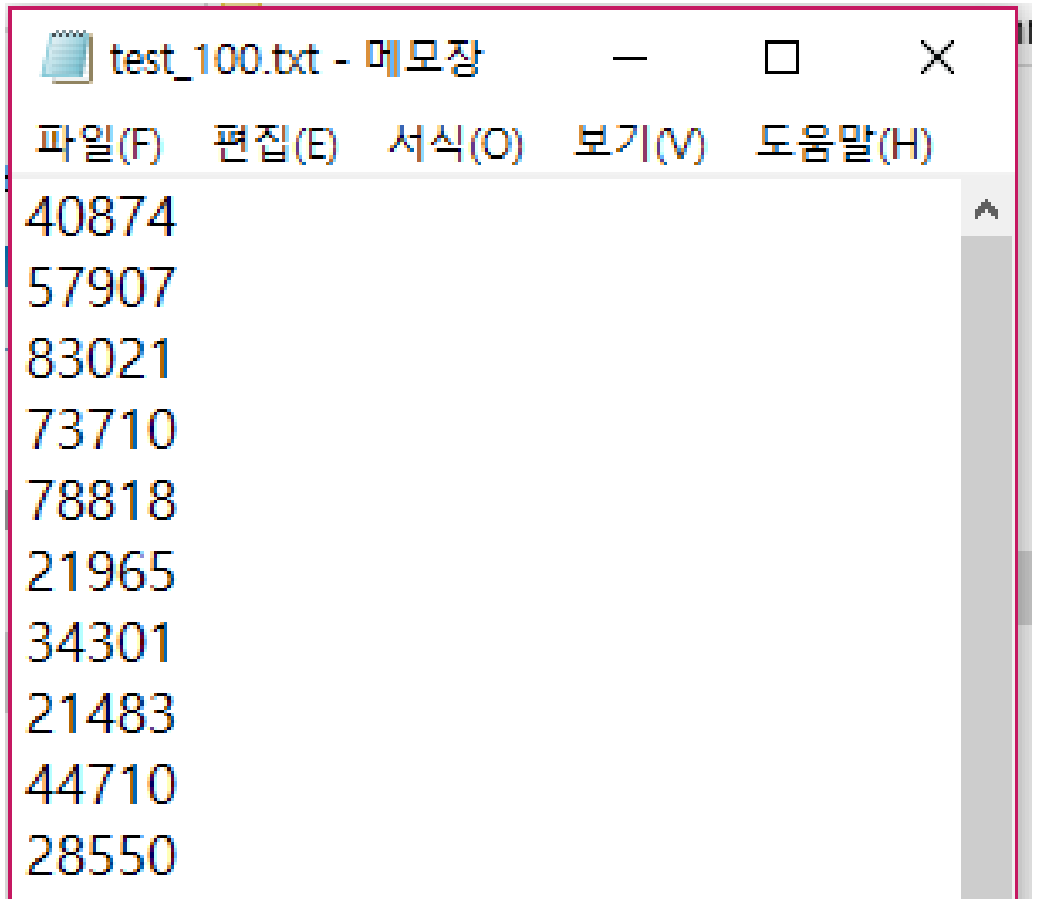
알고리즘 실습 5주차

2018-10-15

과제번호 04


test_100.txt - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
40874
57907
83021
73710
78818
21965
34301
21483
44710
28550

입출력파일 양식

이진 탐색 트리(Binary Search Tree)

- 특정 노드가 가지는 좌측 자식은 항상 해당 노드의 데이터보다 작은 데이터를 가지고 우측 자식은 큰 데이터를 가지는 트리
- Insert, Median Insert
- Recursive Search, Iterative Search
- Successor, Predecessor
- Delete
- 파일 입출력(중위 순회)

이진 탐색 트리(Binary Search Tree) Insert, Median Insert

- 이진 탐색 트리에 데이터를 추가하는 함수
- Insert: 주어진 순서대로 데이터를 이진 탐색 트리에 추가
- Median Insert: 주어진 데이터를 정렬한 뒤 그 중간 값부터 순서대로 이진 탐색 트리에 추가

이진 탐색 트리(Binary Search Tree) Insert, Median Insert

```
TREE-INSERT(T, z)
 1  y = NIL
 2  x = T.root
 3  while x ≠ NIL
 4        y = x
 5        if z.key < x.key
 6              x = x.left
 7        else x = x.right
 8  z.p = y
 9  if y == NIL
10        T.root = z          // tree T was empty
11  elseif z.key < y.key
12        y.left = z
13  else y.right = z
```

이진 탐색 트리(Binary Search Tree) Recursive Search, Iterative Search

- 이진 탐색 트리에서 주어진 데이터의 존재유무를 반환하는 함수
- Recursive Search: 재귀의 형태로 주어진 원소의 존재유무를 반환
- Iterative Search: 일반적인 반복문의 형태로 주어진 원소의 존재유무를 반환

이진 탐색 트리(Binary Search Tree)
Recursive Search

```
Tree-Search(x, k)
      if (x = NULL  or  k = key[x])
           return x;
      if (k < key[x])
           return Tree-Search(left[x], k);
      else
           return Tree-Search(right[x], k);
```

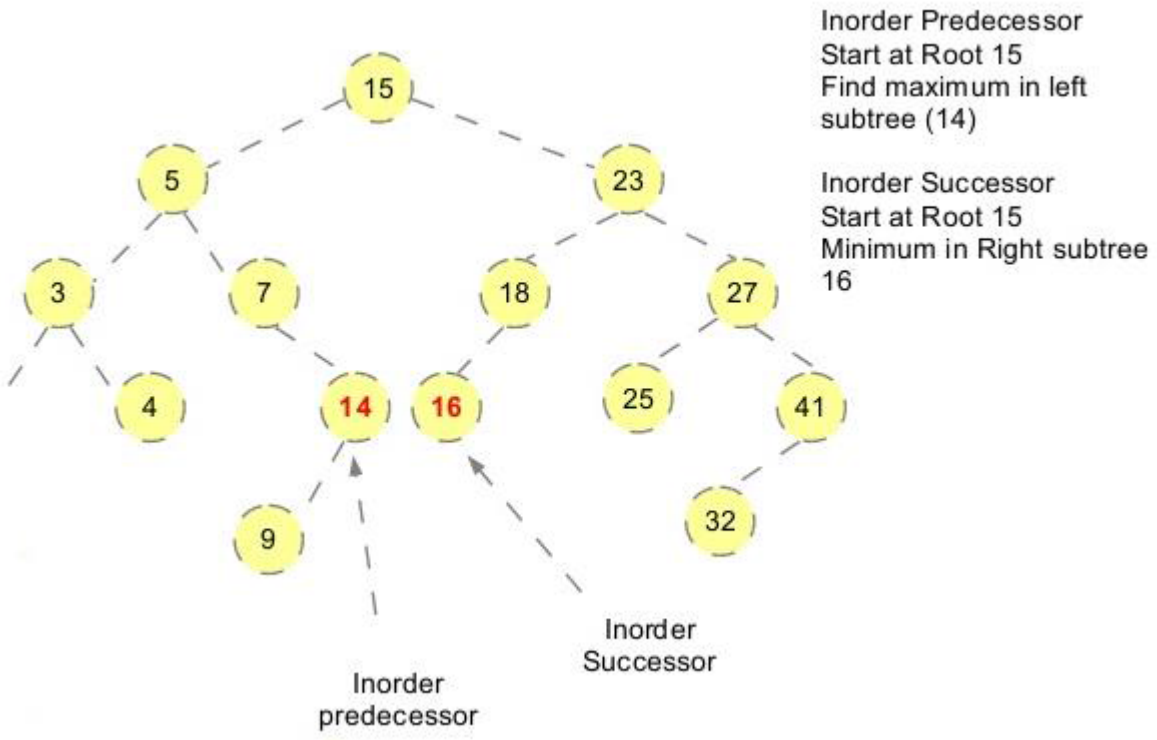
이진 탐색 트리(Binary Search Tree)
Iterative Search

```
Tree-Search(x, k)
    while (x != NULL  and  k != key[x])
        if (k < key[x])
            x = left[x];
        else
            x = right[x];
    return x;
```

이진 탐색 트리(Binary Search Tree) Successor, Predecessor

- 이진 탐색 트리에서 후임(좌측 서브 트리에서 가장 큰 데이터를 가진 노드), 전임자(우측 서브 트리에서 가장 작은 데이터를 가진 노드)를 반환하는 함수들
- Successor: 주어진 노드의 후임을 찾아 반환
- Predecessor: 주어진 노드의 전임자를 찾아 반환

이진 탐색 트리(Binary Search Tree) Successor, Predecessor



이진 탐색 트리(Binary Search Tree) Successor, Predecessor

```
TREE-SUCCESSOR(x)
1  if x.right ≠ NIL
2        return TREE-MINIMUM(x.right)
3  y = x.p
4  while y ≠ NIL and x == y.right
5        x = y
6        y = y.p
7  return y
```

이진 탐색 트리(Binary Search Tree) Delete

- 이진 탐색 트리에서 구조를 깨지 않고 특정 원소를 지우는 함수
- 이진 탐색 트리의 구조(현재 노드를 기준으로 좌측 노드는 작은 데이터 값을 우측 노드는 큰 데이터 값을 가지는 형태)
- 3가지 유형

1. 지우고자 하는 노드가 자식이 없을 때
   I. 지우고자 하는 노드를 지운다.
2. 지우고자 하는 노드가 하나의 자식을 가질 때
   I. 지우고자 하는 노드의 자식을 지우고자 하는 노드의 위치로 이동한다
3. 지우고자 하는 노드가 둘의 자식을 가질 때
   I. 지우고자 하는 노드의 후임과 교체한다.
   II. 위의 다른 유형을 적용하여 지우고자 하는 노드를 지운다.

이진 탐색 트리(Binary Search Tree) Delete

```
TREE-DELETE(T, z)
 1  if z.left == NIL
 2      TRANSPLANT(T, z, z.right)
 3  elseif z.right == NIL
 4      TRANSPLANT(T, z, z.left)
 5  else y = TREE-MINIMUM(z.right)
 6      if y.p ≠ z
 7          TRANSPLANT(T, y, y.right)
 8          y.right = z.right
 9          y.right.p = y
10      TRANSPLANT(T, z, y)
11      y.left = z.left
12      y.left.p = y
```

```
TRANSPLANT(T, u, v)
1  if u.p == NIL
2      T.root = v
3  elseif u == u.p.left
4      u.p.left = v
5  else u.p.right = v
6  if v ≠ NIL
7      v.p = u.p
```

과제 제출

- 사이버캠퍼스를 통해 제출
- 제출 내용: .zip, 보고서
- 파일 이름 양식: [AI]과제번호_학번_이름_언어(C or Java)

EX)[AI]01_201800000_홍길동_Java

- 제출기한: 실습 일로부터 1주일

EX) 9월 10일 18시 실습 수업 시 9월 17일 17시 59분까지 제출

- 추가 제출 감점: 24시간 당 10% 감점 최대 50%감점

EX)9월 10일 18시 실습 수업 시 9월 22일 17시 59분까지 추가제출 가능

질문

- 조교 메일 : 201850855@o.cnu.ac.kr
- 메일 제목(주의): [AI]제목을 입력하세요
- 과제에 대한 수업내용이나 과제 해답에 대한 질문, 언어에 대한 질문 X
- 과제 자체에 대한 이해, 환경설정과 관련된 질문 O
- EX) 출력 파일 이름은 어떤 걸로 해도 상관없나요?, 이런 부분은 조금 애매한 것 같은데 정확히 이런 의도가 맞나요?