

자료구조 (Data Structure)

12주차: 이진 탐색 트리

이진 트리를 포인터로 구현하기 정리

- 이진 트리 노드 타입

- left: 왼쪽 자식 주소
- right: 오른쪽 자식 주소
- value: 저장된 데이터
- /* - parent: 부모 주소 */

이진 트리의 노드 방문하기 정리

함수 VISIT (노드 node):

- PREORDER(node->value)
- VISIT(node->left)
- INORDER(node->value)
- VISIT(node->right)
- POSTORDER(node->value)

특별한 속성을 가진 이진 트리

- 부모 노드와 자식 노드 간에 특정 관계가 항상 성립
 - 최대 힙: 부모 노드가 제일 크다
 - 최소 힙: 부모 노드가 제일 작다
 - 이진탐색트리: 왼쪽 자식 $<$ 부모 $<$ 오른쪽 자식

이번 시간 목차

- 이진 탐색 트리의 특징
- 이진 탐색 트리 예제
- 이진 탐색 트리에 데이터 추가 / 삭제

이진 탐색 트리의 특징 - 현재 노드 관점

- 현재 노드에서 자식 노드들과 데이터를 비교했을 때, 특징은?

이진 탐색 트리의 특징 - left 관점

- 왼쪽 자식 노드에서 봤을때의 특징은?

이진 탐색 트리의 특징 - right 관점

- 오른쪽 자식 노드에서 봤을때의 특징은?

이진 탐색 트리의 특징 - 최소값

- 최소값 노드의 위치는?

이진 탐색 트리의 특징 - 최대값

- 최대값 노드의 위치는?

이진 탐색 트리의 특징 - 왼쪽 서브트리

- 왼쪽 자식이 있을 때, 현재 데이터보다 더 작은 값 중 가장 가까운 값의 위치는?

이진 탐색 트리의 특징 - 왼쪽 조상

- 왼쪽 자식이 없을 때, 현재 데이터보다 더 작은 값 중 가장 가까운 값의 위치는?

이진 탐색 트리의 특징 - 오른쪽 서브트리

- 오른쪽 자식이 있을 때, 현재 데이터보다 더 큰 값 중 가장 가까운 값의 위치는?

이진 탐색 트리의 특징 - 오른쪽 조상

- 오른쪽 자식이 없을 때, 현재 데이터보다 더 큰 값 중 가장 가까운 값의 위치는?

이진 탐색 트리의 특징 - 정렬

- 이진 탐색 트리의 값들을 정렬된 순서로 리스트로 만드는 방법은?

이진 탐색 트리의 특징 - 만들기

- 정렬된 리스트로부터 균형 잡힌 이진 탐색 트리를 만드는 방법은?

이진 탐색 트리의 특징 - 탐색

- 이진 탐색 트리에서 특정 값에 해당하는 노드를 찾는 방법은?

이진 탐색 트리의 특징 - 탐색

함수 SEARCH(노드 n , 찾을 데이터 k):

- 만약: n 이 NULL이면,
반환: NULL
- 만약: $k == n \rightarrow \text{data}$ 면,
반환: n
- 만약: $k < n \rightarrow \text{data}$ 면,
반환: SEARCH($n \rightarrow \text{left}$, k)
- 만약: $k > n \rightarrow \text{data}$ 면,
반환: SEARCH($n \rightarrow \text{right}$, k)

예제 - 구간 트리

- 구간: [low, high]
- $[l1, h1], [l2, h2]$ 가 겹치지 않는다
 - $h1 < l2$
 - $h2 < l1$
- $[l1, h1], [l2, h2]$ 가 겹치지 않는다
 - 그 외의 경우
- 마우스 클릭으로 도형 선택하기 (가로, 세로)

예제 - 구간 트리

- 이진 탐색 트리에서 비교할 기준: low
- 추가로 저장할 데이터: max (자손 포함 최대값)

예제 - 구간 트리

- 어떤 좌표 x 가 현재 노드 $node$ 의 구간에 속했는지
확인하는 방법은?

예제 - 구간 트리

- 어떤 좌표 x 가 현재 노드와 모든 자손에 속하지 않을 조건은?

예제 - 구간 트리

- 어떤 좌표 x 가 현재 노드와 오른쪽 자손에 속하지 않을 조건은?

예제 - 구간 트리

- 어떤 좌표 x 가 왼쪽 자손에 속하지 않을 조건은?

예제 - 구간 트리

함수 SEARCH_ALL(좌표 x , 노드 n):

- 만약: $n == \text{NULL}$ 또는 $(x \rightarrow \text{low}) > (n \rightarrow \text{max})$ 면,
함수 종료
- 만약: $n \rightarrow \text{left} \neq \text{NULL}$ 이고 $(x \rightarrow \text{high}) \leq (n \rightarrow \text{left} \rightarrow \text{max})$,
SEARCH_ALL(x , $n \rightarrow \text{left}$)
- 만약: x 가 n 의 범위에 속하면,
 n 을 보고하기
- 만약: $(x \rightarrow \text{high}) \geq (n \rightarrow \text{low})$ 면,
SEARCH_ALL(x , $n \rightarrow \text{right}$)

이진 탐색 트리 - 노드 추가

- 적절한 위치를 찾아 끝까지 가서 노드를 추가한다

이진 탐색 트리 - 노드 추가

- 노드 추가 함수를 실행한 후에 루트가 반환될 때,
비어있는 트리에 새로운 노드를 추가하면, 반환값은?

이진 탐색 트리 - 노드 추가

- 노드 추가 함수를 실행한 후에 루트가 반환될 때,
존재하는 트리에 새로운 노드를 추가하면, 반환값은?

이진 탐색 트리 - 노드 추가

- 기존 노드와 새로운 노드의 데이터가 같으면?

이진 탐색 트리 - 노드 추가

- 기존 노드의 데이터보다 새로운 노드의 데이터가 작으면?

이진 탐색 트리 - 노드 추가

- 기존 노드의 데이터보다 새로운 노드의 데이터가 크면?

이진 탐색 트리 - 노드 추가

함수 INSERT(루트 노드 node, 새 노드 new_node):

- 만약: `node == NULL`이면,
 반환: `new_node`
- 만약: `(new_node->key) < (node->key)`이면,
 `node->left = INSERT(node->left, new_node)`
- 만약: `(new_node->key) > (node->key)`이면,
 `node->right = INSERT(node->right, new_node)`
- 반환: `node`

이진 탐색 트리 - 노드 삭제

- 노드가 삭제되면, 필요한 경우 트리 구조를 바꿔서
왼쪽 데이터 < 데이터, 데이터 < 오른쪽 데이터를
항상 유지한다

이진 탐색 트리 - 노드 삭제

- 자식이 없는 노드를 삭제할 때, 삭제될 노드의 자리를 대체할 노드는?

이진 탐색 트리 - 노드 삭제

- 자식이 한개인 노드를 삭제할 때, 자리를 대체할 노드는?

이진 탐색 트리 - 노드 삭제

- 자식이 두개인 노드를 삭제할 때, 자리를

대체할 노드는? 그 노드의 자식이 두개일 경우는?

이진 탐색 트리 - 노드 삭제

/* parent 초기값은 NULL */

함수 SET_PARENT(노드 node):

- 만약: node->left != NULL이면,
node->left->parent = node
SET_PARENT(node->left)
- 만약: node->right != NULL이면,
node->right->parent = node
SET_PARENT(node->right)

이진 탐색 트리 - 노드 삭제

함수 EXTRACT(노드 node):

- 만약: node 자식이 없으면,
node->parent의 해당 자식 = NULL
- 그외 만약: node 자식이 하나면,
node->parent의 해당 자식 = node 자식
- 그외:
node->parent의 해당 자식 = node의 다음 노드 s
s->parent->right = s->right

이번 시간 정리

- 이진 탐색 트리의 특징
- 이진 탐색 트리에 노드 추가하기
- 이진 탐색 트리에서 노드 삭제하기

2021 기말고사 기출문제

- 아래의 순서로 data가 입력된다고 할 때 만들어지는 BST를 그림으로 설명하시오.

11 22 33 44 55 66 77

2018 기말고사 기출문제

- Binary search tree에 1부터 100까지의 숫자가 저장되어 있다고 가정하자. 이 binary search tree에서 55를 탐색하는 경우 방문되는 node에 저장된 값을 적었을 때, 다음 중 발생할 수 없는 sequence는 어느 것인지 고르고, 그 이유를 적으시오.

- ① {10, 75, 64, 43, 60, 57, 55}
- ② {90, 12, 68, 34, 62, 45, 55}
- ③ {9, 85, 47, 68, 43, 57, 55}
- ④ {79, 14, 72, 56, 16, 53, 55}

2018 기말고사 기출문제

- 어떤 binary search tree의 preorder traversal sequence가 30, 20, 10, 15, 25, 23, 39, 35, 42일 때, 이 binary search tree를 그림으로 나타내시오. Binary search tree를 확정하는 과정을 보이시오.