

자료구조 (Data Structure)

6주차: 힙

이진 트리를 포인터로 구현하기 정리

- 이진 트리 노드 타입
 - left: 왼쪽 자식 주소
 - right: 오른쪽 자식 주소
 - value: 저장된 데이터

이번 시간 목차

- 이진 트리를 배열로 구현하기
- 특별한 속성을 가진 이진 트리
- 특별한 속성을 가진 이진 트리 예제

이진 트리를 배열로 구현하기 - 배열

- 배열의 특징은?

이진 트리를 배열로 구현하기 - 저장 순서

- 배열의 첫번째에 저장하기에 적합한 데이터는?
- 배열의 두번째에 저장하기에 적합한 데이터는?

이진 트리를 배열로 구현하기 - 순서 계산

- 자식 데이터의 순서를 계산하는 방법은?
- 부모 데이터의 순서를 계산하는 방법은?

이진 트리를 배열로 구현하기 - 추가, 삭제

- 데이터를 추가할 적절한 위치는?
- 데이터를 삭제할 때의 적절한 동작은?

이진 트리를 배열로 구현하기 - 구조체

- 배열의 주소와 크기 정보 저장
 - data: 배열 주소
 - capacity: 배열 크기
 - size: 현재 저장된 노드 개수

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

특별한 속성을 가진 이진 트리 - 최대힙

- 최대 힙에 데이터를 추가할 적절한 위치는?

특별한 속성을 가진 이진 트리 - 최대힙

함수 MAX_HEAP_INSERT(heap, value):

- 실행조건: $\text{heap} \rightarrow \text{size} < \text{heap} \rightarrow \text{capacity}$
- $\text{++heap} \rightarrow \text{size}$
- $\text{heap} \rightarrow \text{data}[\text{heap} \rightarrow \text{size}] = \text{value}$
- HEAPIFY_UP(heap, $\text{heap} \rightarrow \text{size}$)

특별한 속성을 가진 이진 트리 - 최대힙

- 최대 힙의 끝에 데이터가 추가된 경우,
다시 힙 속성을 회복하려면?

특별한 속성을 가진 이진 트리 - 최대힙

함수 HEAPIFY_UP(heap, child):

- 만약: child == 0이면 반환
- parent = child / 2
- 만약: heap->data[parent] < heap->data[child]이면
 tmp = heap->data[parent]
 heap->data[parent] = heap->data[child]
 heap->data[child] = tmp
 HEAPIFY_UP(heap, parent)

특별한 속성을 가진 이진 트리 - 최대힙

- 새로운 최대 힙에 5, 3, 8, 1, 2를 순서대로 추가하면?

특별한 속성을 가진 이진 트리 - 최대힙

- 최대 힙에서 데이터를 삭제한 경우, 그 자리를 메꿀 노드는?

특별한 속성을 가진 이진 트리 - 최대힙

함수 MAX_HEAP_DELETE(heap, index):

- 실행조건: $\text{heap} \rightarrow \text{size} > 0$
- $\text{value} = \text{heap} \rightarrow \text{data}[\text{index}]$
- $\text{heap} \rightarrow \text{data}[\text{index}] = \text{heap} \rightarrow \text{data}[\text{heap} \rightarrow \text{size}]$
- $\text{heap} \rightarrow \text{size}--$
- HEAPIFY_DOWN(heap, index)
- 반환: value

특별한 속성을 가진 이진 트리 - 최대힙

- 최대 힙에서 삭제된 데이터 자리를 제일 끝 노드가 메꿨다면, 힙 속성을 회복할 방법은?

특별한 속성을 가진 이진 트리 - 최대힙

함수 HEAPIFY_DOWN(heap, parent):

- left = parent * 2
- right = parent * 2 + 1
- largest = parent
- 만약: $\text{left} \leq \text{heap->size}$
 $\&& \text{heap->data[left]} > \text{heap->data[largest]}$ 이면
largest = left
- 만약: $\text{right} \leq \text{heap->size}$
 $\&& \text{heap->data[right]} > \text{heap->data[largest]}$ 이면
largest = right

특별한 속성을 가진 이진 트리 - 최대힙

- 만약: $\text{largest} \neq \text{parent}$ 이면

```
tmp = heap->data[parent]
```

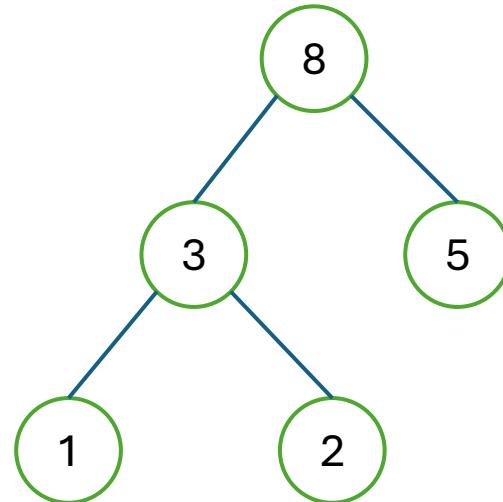
```
heap->data[parent] = heap->data[largest]
```

```
heap->data[largest] = tmp
```

```
HEAPIFY_DOWN(heap, largest)
```

특별한 속성을 가진 이진 트리 - 최대힙

- 다음 최대 힙에서 루트 노드를 삭제하면?



예제 - 우선순위 큐

- 우선순위가 높은 데이터가 먼저 삭제되는 큐
 - 최대 힙을 이용하여 구현한다
 - 추가: 최대 힙에 데이터 추가
 - 삭제: 최대 힙에서 루트 노드를 제거

예제 - 소셜미디어 데이터

- 소셜미디어 홈페이지에서 관리해야 할 자료는?

예제 - 소셜미디어 데이터

- 각 유저마다 저장할 정보는?

예제 - 소셜미디어 데이터

- 유저 데이터 구조체
 - 친구 유저 리스트
 - 게시물 리스트 (최신순으로 정렬됨)

예제 - 소셜미디어 데이터 타임라인

- 유저의 최신 게시물과 친구들의 최신 게시물들을
통틀어 가장 최신인 10개 게시물 리스트를 만들려면?

예제 - 소셜미디어 데이터 타임라인

함수 GET_TIMELINE(user):

- result = 새로운 게시물 리스트
- Q = 새로운 게시물 노드 우선순위 큐
- 반복: user의 모든 친구 friend에 대해
만약: friend의 게시물이 있으면
ENQUEUE(Q, friend의 게시물들의 first)

예제 - 소셜미디어 데이터 타임라인

- 반복: $\text{SIZE}(Q) > 0$ 그리고 $\text{SIZE}(\text{result}) < 10$
 $\text{node} = \text{DEQUEUE}(Q)$
 $\text{PUSH}(\text{result}, \text{node}->\text{data})$
 만약: $\text{node}->\text{next} \neq \text{NULL}$ 이면
 $\text{ENQUEUE}(Q, \text{node}->\text{next})$
- 반환: result

예제 - 소셜미디어 데이터

참고 자료

- <https://leetcode.com/problems/design-twitter/>

이번 시간 정리

- 이진 트리를 배열로 구현하기
- 힙의 정의와 데이터 추가, 삭제
- 힙을 이용한 우선순위 큐 예제를 살펴보았다

2021 중간고사 기출문제 (1 / 14)

- 아주대학교 재학생들의 소프트웨어학과 개설 과목
이수 여부를 저장할 때 자료 구조를 추천하시오

2021 중간고사 기출문제 (2 / 14)

- 매일 행적을 기록할 때 자료 구조를 추천하시오

2021 중간고사 기출문제 (3 / 14)

- 식당에서 손님들의 주문 내역을 기록할 때 쓸 자료구조를 추천하시오

2021 중간고사 기출문제 (4 / 14)

- 여행 계획을 세울 때, 여러 상황 및 대응 계획을 저장할 자료구조를 추천하시오 (다음 상황 중복 없음)

2021 중간고사 기출문제 (5 / 14)

- 쇼핑할 목록이 자주 바뀐다고 가정했을 때,
쇼핑할 목록을 저장할 자료구조를 추천하시오

2021 중간고사 기출문제 (6 / 14)

- Stack에 데이터를 추가, 삭제할 때 드는 시간은?

2021 중간고사 기출문제 (7 / 14)

- Queue에 데이터를 추가, 삭제할 때 드는 시간은?

2021 중간고사 기출문제 (8 / 14)

- Singly linked list의 시작 node 주소만 알고 있을 때, 특정 값을 가진 node를 삭제할 때 드는 시간은?

2021 중간고사 기출문제 (9 / 14)

- Doubly linked list의 head node의 주소만 알 때, 마지막 노드를 삭제하는데 드는 시간은?

2021 중간고사 기출문제 (10 / 14)

- Binary Tree에서 i 층에 있는 노드들의 data를 출력하는데 드는 시간은?

2021 중간고사 기출문제 (11 / 14)

- chain 형태의 linked list의 데이터들을 리스트 내에 저장되어 있는 순서와 반대로 출력하는 재귀 함수는?

2021 중간고사 기출문제 (12 / 14)

- 1 + 2 형태로 표현된 수식을 1 2 + 형태로 변환하는 코드를 쓰시오

2021 중간고사 기출문제 (13 / 14)

- chain 형태의 linked list에 노드를 추가하는 코드를 쓰시오

2021 중간고사 기출문제 (14 / 14)

- inorder traversal 결과가 CBEDFAIHJG이고 postorder traversal 결과가 ABCD EFGHIJ인 binary tree를 그림으로 표현하시오