힙?

201411802 응용통계학과 전민규

노드들이 저장하고 있는 키들이 다음과 같은 식을 만족하는 완전 이진트리

최대 힙(Max heap)

-부모 노드의 키값이 자식 노드의 키값보다 크거나 같은 완전 이진트리

-key(부모노드) >= key(자식노드)

최소 힙(Min Heap)

-부모 노드의 키값이 자식 노드의 키값보다 작거나 같은 완전 이진트리

-key(부모노드) <= key(자식노드)

힙의 구현방법

힙은 연결리스트와 배열을 이용하여 구할 수 있다

완전 이진트리이므로 배열을 이용하면 쉽게 구현가능

-완전 이진트리이므로 각 노드에 번호를 붙일 수 있다

-이 번호를 배열의 인덱스라고 생각하자

부모노드와 자식노드를 찾기 쉽다

-왼쪽 자식의 인덱스 = (부모의 인덱스)\*2

-오른쪽 자식의 인덱스 = (부모의 인덱스)\*2+1

-부모의 인덱스=(자식의 인덱스)/2

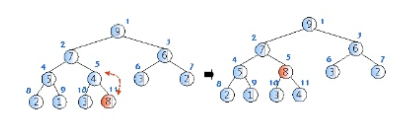
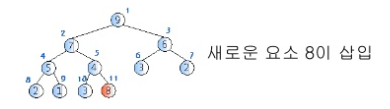
힙에 새로운 요소 삽입

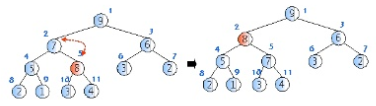
-힙에 새로운 요소가 들어오면, 일단 새로운 노드를 힙의 마지막 노드 다음에 삽입

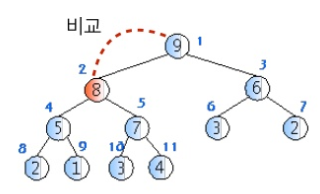
-삽입된 새로운 노드를 부모 노드들과 교환해서 힙의 성질을 만족하도록 함(교환연산)

-위 과정을 힙의 성질이 만족될 때 까지 반복

-> 이 과정을 upheap







8과 부모노드 9를 비교하니 자식 노드가 부모 노드보다 작은값 이다. 이제 힙의 성질을 만족하므로 더 이상 교환하지 않는다.

노드의 삭제

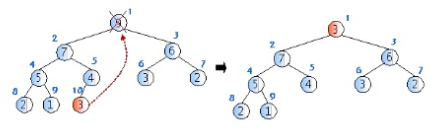
최대 힙에서의 삭제는 가장 큰 키값을 가진 노드(root node)를 삭제하는 것을 의미

-root 노드를 삭제한다

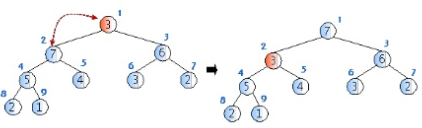
-마지막 노드를 root 노드로 이동한다

-루트노드에서부터 단말 노드까지의 경로에 있는 노드들을 교환하여 힙 성질을 만족시키다.

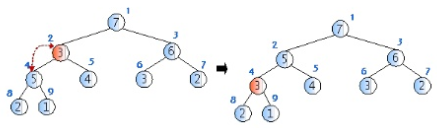
->위 과정을 down heap이라 한다.



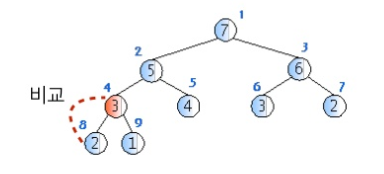
루트노드(최댓값) 9를 삭제하고 제일 마지막 노드3을 루트노드로 만듦



루트노드 3과 왼쪽 자식노드 7을 비교하여 heap의 성질을 만족하도록 한다(7과 3을 교환)



교환된 3과 왼쪽 자식노드 5를 비교하여 heap의 성질을 만족하도록 한다(3과 5를 교환)



3과 그 자식노드 2를 비교하니 자식 노드가 부모 노드보다 작은 값이다.

이제 힙의 성질을 만족하므로 더 이상 교환이 없음

힙의 복잡도 분석

삽입 연산에서 최악의 경우, 루트 노드까지 올라가야 하므로 트리의 높이에 해당하는 비교 연산 및 이동 연산이 필요->O(logN)

삭제도 최악의 경우, 가장 아래 레벨까지 내려가야 하므로 역시 트리의 높이 만큼의 시간이 걸린다.->O(logN)

힙정렬

먼저 정렬해야 할 N개의 요소들을 최대 힙에 삽입

한번에 하나씩 요소를 힙에서 삭제하여 나오는 값을 저장하면 된다(루트노드(최대값)부터 삭제되면서 출력되므로)

하나의 요소를 힙에 삽입하거나 삭제할 때 시간이 O(logN)만큼 소요되고 요소의 개수가 N개 이므로 전체적으로 O(NlogN)시간이 걸린다.

힙 정렬이 가장 유용한 것은 전체 자료를 정렬하는 것이 아니라 가장 큰 값 몇 개만 필요할 때이다.