

알고리즘

과제번호	02주차
날 짜	2018.09.29
학 번	201302395
이 름	류경빈

실행시간 분석

INSERT - 삽입

삽입이 실행되기 위해서는 MAX-HEAP의 가장 마지막 key에 새롭게 추가해야한다.

이럴 경우, 새로운 노드를 $\text{leaf}(\text{heap-size})$ 위치에 추가 한 후 해당 위치에서부터 parent 노드와 계속 비교를 실행해 MAX-HEAP의 조건에 알맞게 구성이 될 때까지 반복해서 비교한다 (maxHeapInsert). 그래서 시간 복잡도는 트리의 높이에 비례하게 되기 때문에 $O(\log n)$ 이다.

MAX - 최대값

MAX-HEAP의 최대값을 반환하는 $\text{max}(S)$ 함수는 노드의 1번째 값을 반환하면 가능하다.

이유는 MAX-HEAP의 root는 항상 key값이 최대인 노드를 가지고 있기 때문이다. 그래서 max 의 시간 복잡도는 항상 같은 값을 가지기 때문에 $O(1)$ 이다.

EXTRACT_MAX - 최대값 제거

MAX-HEAP의 최대 key 값은 root 노드이다. EXTRACT_MAX의 알고리즘은 root 값을 leaf값(제일 key값이 낮은 노드)으로 바꿔준 후 heap의 사이즈를 줄여준 후 노드를 다시 BUILD-MAX-HEAP을 통해 MAX-HEAPIFY를 해주는 과정을 진행한다. 그러면 새롭게 root부터 최대 키 값을 가지는 MAX-HEAP 구조로 만들 수 있다. 이럴 경우 시간 복잡도는 MAX-HEAPIFY를 한 번 진행하기 때문에 $O(\log n)$ 이다.

INCREASE_KEY - 원소 값 KEY 증가

입력으로부터 받은 인덱스의 key값을 증가시키기 위해서는 해당 인덱스의 key 값을 수정해준 후 해당 위치부터 MAX-HEAP 조건을 맞추어야 한다. 기존에 parent가 다른 tree들은 이미 맞춰져 있기 때문에 해당 위치의 maxHeapInsert 를 실행해 구현한다. 시간 복잡도는 $O(\log n)$ 이다.

H_DELETE - 노드 제거

입력으로부터 받은 인덱스를 제거하기 위한 알고리즘은 leaf값을 입력받은 인덱스의 값으로 바꿔준 후 heap 사이즈를 줄인다. 그리고 새롭게 MAX-HEAPIFY를 실행해 달라진 HEAP 구조를 MAX-HEAP으로 바꿔준다. 이럴 경우 시간 복잡도는 MAX-HEAPIFY를 한 번 실행해야 하므로 $O(\log n)$ 이다.

결과 화면

**** 현재 우선 순위 큐에 저장되어 있는 작업 대기 목록은 다음과 같습니다. ****

[0] 230, 수치해석
[1] 70, 자료구조 및 실습
[2] 150, 파일처리론
[3] 40, 컴퓨터 구조2
[4] 60, 객체지향설계
[5] 80, 기초물리학
[6] 98, 계산이론
[7] 38, 논리회로 및 실험
[8] 30, 컴퓨터 구조1
[9] 41, 고급프로그래밍설계
[10] 45, 소프트웨어 설계
[11] 56, 소프트웨어 공학
[12] 9, 선형대수
[13] 1, 컴퓨터프로그래밍1
[14] 3, 컴퓨터프로그래밍2
[15] 27, 이산수학
[16] 29, 프로그래밍 언어

1. 작업추가 2. 최대값 3. 최대 우선순위 작업처리
4. 원소 키값 증가 5. 작업 제거 6. 종료

신규 작업명 : **알고리즘**
우선 순위 (0~999) : **160**
**** 작업 추가 완료 ****

[0] 230, 수치해석
[1] 160, 알고리즘
[2] 150, 파일처리론
[3] 70, 자료구조 및 실습
[4] 60, 객체지향설계
[5] 80, 기초물리학
[6] 98, 계산이론
[7] 38, 논리회로 및 실험
[8] 40, 컴퓨터 구조2
[9] 41, 고급프로그래밍설계
[10] 45, 소프트웨어 설계
[11] 56, 소프트웨어 공학
[12] 9, 선형대수
[13] 1, 컴퓨터프로그래밍1
[14] 3, 컴퓨터프로그래밍2
[15] 27, 이산수학
[16] 29, 프로그래밍 언어
[17] 30, 컴퓨터 구조1

[0] 230, 수치해석
[1] 160, 알고리즘
[2] 150, 파일처리론
[3] 70, 자료구조 및 실습
[4] 60, 객체지향설계
[5] 80, 기초물리학
[6] 98, 계산이론
[7] 38, 논리회로 및 실험
[8] 40, 컴퓨터 구조2
[9] 41, 고급프로그래밍설계
[10] 45, 소프트웨어 설계
[11] 56, 소프트웨어 공학
[12] 9, 선형대수
[13] 1, 컴퓨터프로그래밍1
[14] 3, 컴퓨터프로그래밍2
[15] 27, 이산수학
[16] 29, 프로그래밍 언어
[17] 30, 컴퓨터 구조1

1. 작업추가 2. 최대값 3. 최대 우선순위 작업처리
4. 원소 키값 증가 5. 작업 제거 6. 종료

**** MAX : 230, 수치해석****

[0] 300, 알고리즘
[1] 230, 수치해석
[2] 150, 파일처리론
[3] 70, 자료구조 및 실습
[4] 60, 객체지향설계
[5] 80, 기초물리학
[6] 98, 계산이론
[7] 38, 논리회로 및 실험
[8] 40, 컴퓨터 구조2
[9] 41, 고급프로그래밍설계
[10] 45, 소프트웨어 설계
[11] 56, 소프트웨어 공학
[12] 9, 선형대수
[13] 1, 컴퓨터프로그래밍1
[14] 3, 컴퓨터프로그래밍2
[15] 27, 이산수학
[16] 29, 프로그래밍 언어
[17] 30, 컴퓨터 구조1

1. 작업추가 2. 최대값 3. 최대 우선순위 작업처리
4. 원소 키값 증가 5. 작업 제거 6. 종료

**** 한 개의 작업을 처리했습니다. ****

[0] 230, 수치해석
[1] 70, 자료구조 및 실습
[2] 150, 파일처리론
[3] 40, 컴퓨터 구조2
[4] 60, 객체지향설계
[5] 80, 기초물리학
[6] 98, 계산이론
[7] 38, 논리회로 및 실험
[8] 30, 컴퓨터 구조1
[9] 41, 고급프로그래밍설계
[10] 45, 소프트웨어 설계
[11] 56, 소프트웨어 공학
[12] 9, 선형대수
[13] 1, 컴퓨터프로그래밍1
[14] 3, 컴퓨터프로그래밍2
[15] 27, 이산수학
[16] 29, 프로그래밍 언어

수정할 key : **190**
**** 한 개의 작업을 처리했습니다. ****

[0] 230, 수치해석
[1] 70, 자료구조 및 실습
[2] 190, 선형대수
[3] 40, 컴퓨터 구조2
[4] 60, 객체지향설계
[5] 150, 파일처리론
[6] 98, 계산이론
[7] 38, 논리회로 및 실험
[8] 30, 컴퓨터 구조1
[9] 41, 고급프로그래밍설계
[10] 45, 소프트웨어 설계
[11] 56, 소프트웨어 공학
[12] 80, 기초물리학
[13] 1, 컴퓨터프로그래밍1
[14] 3, 컴퓨터프로그래밍2
[15] 27, 이산수학
[16] 29, 프로그래밍 언어

1. 작업추가 2. 최대값 3. 최대 우선순위 작업처리
4. 원소 키값 증가 5. 작업 제거 6. 종료

삭제할 노드 x 입력 : **9**
**** 한 개의 작업을 처리했습니다. ****

[0] 230, 수치해석
[1] 70, 자료구조 및 실습
[2] 190, 선형대수
[3] 40, 컴퓨터 구조2
[4] 45, 소프트웨어 설계
[5] 150, 파일처리론
[6] 98, 계산이론
[7] 38, 논리회로 및 실험
[8] 30, 컴퓨터 구조1
[9] 41, 고급프로그래밍설계
[10] 29, 프로그래밍 언어
[11] 56, 소프트웨어 공학
[12] 80, 기초물리학
[13] 1, 컴퓨터프로그래밍1
[14] 3, 컴퓨터프로그래밍2
[15] 27, 이산수학