

알고리즘 실습 보고서

-MaxPQ-

전공 :컴퓨터공학과

분반 : 05반

학번 :201701988

이름 :김수빈

1. 실행 환경

본 실습은 Windows10 64bit, jdk 1.8.0_221, Eclipse EE가 설치된 환경에서 실행되었다

2. 과제 설명

- 출제된 과제에 대한 설명

이번 과제는 data04.txt 파일을 읽어 Heap을 구성하여 우선순위 큐를 구현하는 것이다.

콘솔 창에서 사용자 입력을 받아 삽입, 삭제, 최대값 확인, 최대값 삭제, 키 값 증가의 연산이 가능하며, 연산 이후에도 우선순위 큐를 유지하도록 구현한다.

3. 문제 해결 방법

- 문제를 해결하기 위해 자신이 사용한 방법, 아이디어에 대한 설명

우선순위 큐를 구현하기 위해 필요한 클래스를 3가지로 정의했다. 먼저 사용자 입력을 받아 그에 맞는 메소드를 호출하는 heap 클래스, ArrayList를 받아 max-heap을 만들고 삽입, 삭제 등의 연산을 할 수 있는 makeHeap 클래스, key값과 name을 갖는 노드를 구성하는 Node 클래스가 있다.

▶ heap class

heap 클래스에서는 main 메소드가 실행이 되는데, 먼저 data04.txt 파일을 한 줄씩 읽어, 노드를 구성한다. 숫자를 키값으로 하고 과목명을 name으로 정의한다. ArrayList nodeList에 구성한 노드를 담는다. nodeList를 매개변수로 넘겨 makeHeap 객체 heapList를 선언하여 max-heap을 구성한다. while문을 통해 사용자가 종료를 입력할 때까지 다음을 반복한다.

heapList의 list의 키값과 과목명을 출력하여 현재 작업대기 목록을 보여준다. 사용자 입력을 받아 작업추가, 최대값, 최대 우선순위 작업 처리, 원소 키값 증가, 작업 제거, 종료를 실행한다.

작업 추가일 경우, 사용자 입력을 받아 노드를 구성해 heapList의 insert 메소드를 호출한다.

최대값일 경우, heapList의 max 메소드 리턴값을 출력한다.

최대 우선순위 작업 처리일 경우, heapList의 extract_max 메소드를 호출한다.

원소 키값 증가일 경우, 원소 이름과 새로운 키값을 입력받아 heapList의 get 메소드를 호출하여 원소 이름과 같은 노드를 불러와 노드와 key값을 매개변수로 heapList의 increase_key 메소드를 호출한다.

작업 제거일 경우, 원소 이름을 입력받아 heapList의 get 메소드를 호출하여 원소 이름과 같은 노드를 불러와 노드를 매개변수로 하여 heapList의 delete 메소드를 호출한다.

종료일 경우, abort 플래그를 true로 세팅하여 while문을 빠져나오도록 한다.

▶ makeHeap class

makeHeap 구조체는 Node를 담는 ArrayList list와 heap_size를 전역변수로 갖는다. makeHeap 생성자와 초기 max-heap을 만드는 build_max_heap 메소드, max_heapify 메소드, 새로운 노드를 삽입하는 insert 메소드, 노드를 삭제하는 delete 메소드, 노드의 키 값을 변경하는 increase_key 메소드, 노드의 인덱스를 탐색하는 indexOf 메소드, 노드의 name을 통해 노드를 탐색하는 get 메소드, list 내 키값이 최대인 노드를 반환하는 max, 키값이 최대인 노드를 삭제하는 extract_max 메소드가 있다.

- makeHeap 생성자는 ArrayList를 매개변수로 받아 list로 저장하고, build_max_heap 메소드

를 호출하여 max-heap을 구성한다.

- build_max_heap 메소드는 ArrayList를 매개변수로 받아 자식을 갖는 마지막 노드의 인덱스부터 0까지 max_heapify 호출을 반복한다.

- max_heapify는 ArrayList와 int i 값을 매개변수로 받아 i의 자식 인덱스를 구하여 heap_size 안에 있는 자식들의 경우, 자신과 자식들의 키값을 비교하여 가장 큰 키값을 갖는 노드가 자신이 아닐 경우, swap 한다. 이때, 인덱스 i와 swap 한 노드의 인덱스 largest에서 다시 max_heapify를 호출한다.

- insert 메소드의 경우, Node node를 매개변수로 받아 ArrayList의 맨 끝에 삽입한 후에, node의 인덱스를 index에 저장한다. while문을 통해 다음을 반복한다. index 위치의 부모노드의 키값과 node의 키값을 비교하여 부모의 키값이 더 작고, index가 0이 아닐 경우, index 위치에 부모 노드를 복사해 넣어놓고, index를 부모노드의 인덱스로 한다. index위치에 node를 복사해 넣고 heap_size를 증가시킨다.

- indexOf 메소드는 Node를 매개변수로 받아 리스트를 선형 탐색하여 Node 클래스 내 same 메소드를 사용하여 입력받은 노드와 리스트의 노드가 같은지 탐색한다. 같으면 해당 인덱스를 반환하고 같은 노드가 없는 경우 -1을 반환한다.

- get 메소드는 String을 매개변수로 받아 입력받은 String으로 새로운 노드를 생성하여 indexOf와 같은 방식으로 탐색한다. 같으면 해당 노드를 반환한다. 같은 노드가 없는 경우 null을 반환한다.

- max 메소드는 list의 인덱스 0의 노드를 반환한다.

- extract_max 메소드의 경우 delete 메소드를 호출하여 인덱스 0의 노드를 삭제한다.

- increase_key 메소드의 경우, indexOf 메소드를 통해 node의 list 내 인덱스 값을 index에 저장해놓고, node.name과 입력받은 key로 새로운 노드를 생성한다. list 내 index 위치를 생성한 노드로 바꾼다. while문을 통해 현재 위치 index부터 시작하여 부모노드의 키값과 key를 비교하여 key가 더 크면, list 내 현재 위치의 노드를 부모노드를 복사해 붙여 넣고 index를 부모노드의 인덱스로 바꾸어, 부모노드의 부모노드로 타고 올라간다. key값이 부모노드의 키값보다 작거나 같은 경우 또는 index가 0일 경우 while문을 빠져나온다. list 내 index 위치의 노드를 새로 생성한 노드로 한다.

- delete 메소드는 먼저 indexOf 메소드를 통해 매개변수로 입력한 노드의 인덱스를 index에 저장하고, heap_size를 줄이고 ArrayList의 맨 마지막 노드를 index 위치로 바꾼다. index 위치부터 max_heapify를 호출하여 힙화 한다.

▶ Node class

Node 클래스는 int형 key와 String name을 전역변수로 갖는다. 생성자는 key값과 name을 모

두 매개변수로 받는 생성자와 name만 매개변수로 받는 생성자가 있다. 매개변수로 입력받은 값을 this.key값과 this.name에 저장한다.(name만 입력받는 경우는 key값이 0이다.)

main 메소드에서 우선 순위 큐를 출력할 때 주어진 예시처럼 출력하기 위해 toString 메소드를 사용하여 key+", "+name을 리턴해, Node를 print할 때 key, name처럼 출력되도록 한다.

두 노드가 같은지는 name 값을 비교해 판단한다. 두 노드의 name이 equal이면 true를 리턴하고 아니면 false를 리턴한다.

4. 결과 화면

1. 작업 추가 - 컴퓨터그래픽스, 109

```
Console
heap [Java Application] C:\Program Files\Java\jdk1.8.0_221\bin\javaw.exe (2019. 1
**** 현재 우선 순위 큐에 저장되어 있는 작업 대기 목록은 다음과 같습니다. ****
17개 입니다.

230, 수치해석
70, 자료구조 및 실습
150, 파일처리론
40, 컴퓨터 구조2
60, 객체지향설계
80, 기초물리학
98, 계산이론
38, 논리회로 및 실험
30, 컴퓨터 구조1
41, 고급프로그램설계
45, 소프트웨어 설계
56, 소프트웨어 공학
9, 선형대수
1, 컴퓨터프로그래밍1
3, 컴퓨터프로그래밍2
27, 이산수학
29, 프로그래밍 언어

-----
1. 작업 추가  2. 최대값  3. 최대 우선순위 작업 처리
4. 원소 키값 증가  5. 작업 제거  6. 종료
-----

1
원소 이름과 키 값을 입력하세요.
(ex) '컴퓨터그래픽스, 109'
컴퓨터그래픽스, 109

**** 현재 우선 순위 큐에 저장되어 있는 작업 대기 목록은 다음과 같습니다. ****
18개 입니다.

230, 수치해석
109, 컴퓨터그래픽스
150, 파일처리론
70, 자료구조 및 실습
60, 객체지향설계
80, 기초물리학
98, 계산이론
38, 논리회로 및 실험
40, 컴퓨터 구조2
41, 고급프로그램설계
45, 소프트웨어 설계
56, 소프트웨어 공학
9, 선형대수
1, 컴퓨터프로그래밍1
3, 컴퓨터프로그래밍2
27, 이산수학
29, 프로그래밍 언어
30, 컴퓨터 구조1
```

2. 최대값

2

최대 우선 순위를 갖는 작업입니다.

230, 수치해석

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

18개 입니다.

230, 수치해석

109, 컴퓨터그래픽스

150, 파일처리론

70, 자료구조 및 실습

60, 객체지향설계

80, 기초물리학

98, 계산이론

38, 논리회로 및 실험

40, 컴퓨터 구조2

41, 고급프로그래밍설계

45, 소프트웨어 설계

56, 소프트웨어 공학

9, 선형대수

1, 컴퓨터프로그래밍1

3, 컴퓨터프로그래밍2

27, 이산수학

29, 프로그래밍 언어

30, 컴퓨터 구조1

3. 최대 우선순위 작업 처리

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

3

**** 현재 우선 순위 큐에 저장되어 있는 작업 대기 목록은 다음과 같습니다. ****
17개 입니다.

150, 파일처리론
109, 컴퓨터그래픽스
98, 계산이론
70, 자료구조 및 실습
60, 객체지향설계
80, 기초물리학
30, 컴퓨터 구조1
38, 논리회로 및 실험
40, 컴퓨터 구조2
41, 고급프로그래밍설계
45, 소프트웨어 설계
56, 소프트웨어 공학
9, 선형대수
1, 컴퓨터프로그래밍1
3, 컴퓨터프로그래밍2
27, 이산수학
29, 프로그래밍 언어

4. 원소 키 값 증가

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

4

원소 이름과 새로운 키 값을 입력하세요. (새로이 입력된 키 값은 기존 키 값보다 커야 합니다.)

(ex) '프로그래밍언어개론, 159'

객체지향설계, 99

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

17개 입니다.

150, 파일처리론
109, 컴퓨터그래픽스
98, 계산이론
70, 자료구조 및 실습|
99, 객체지향설계
80, 기초물리학
30, 컴퓨터 구조1
38, 논리회로 및 실험
40, 컴퓨터 구조2
41, 고급프로그램설계
45, 소프트웨어 설계
56, 소프트웨어 공학
9, 선형대수
1, 컴퓨터프로그래밍1
3, 컴퓨터프로그래밍2
27, 이산수학
29, 프로그래밍 언어

5. 작업 제거 & 종료

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

5

원소 이름을 입력하세요.

컴퓨터 구조1

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

16개 입니다.

150, 파일처리론
109, 컴퓨터그래픽스
98, 계산이론
70, 자료구조 및 실습
99, 객체지향설계
80, 기초물리학
29, 프로그래밍 언어
38, 논리회로 및 실험
40, 컴퓨터 구조2
41, 고급프로그램설계
45, 소프트웨어 설계
56, 소프트웨어 공학
9, 선형대수
1, 컴퓨터프로그래밍1
3, 컴퓨터프로그래밍2
27, 이산수학

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

6

|

<

5. 느낀점 및 고찰

구현하는 데에 큰 어려움은 없었습니다. 하지만 실습 수업 없이 과제가 나와 과제 조건을 명확히 이해하기에 어려움이 있었습니다.

6. 이론 시간에 출제된 과제

Build - Max - Heap (A) Analysis

왜 시간비용이 $O(n)$ 에 수렴하는가?

$$\frac{n}{4}(1c) + \frac{n}{8}(2c) + \frac{n}{16}(3c) + \dots + 1(\lg n c) \quad c \text{는 constant 비용}$$

$\frac{n}{4} = 2^k$ 라 하자.

$$c \times 2^k \left(\frac{1}{2^0} + \frac{2}{2^1} + \frac{3}{2^2} + \frac{4}{2^3} + \dots + \frac{(k+1)}{2^k} \right)$$
$$= c \times 2^k \times \sum_{x=1}^{k+1} \frac{x}{2^{x-1}} = c \times 2^k \times 4$$
$$= c \times \frac{n}{4} \times 4$$
$$= cn$$

c 는 constant 비용이므로 $\therefore n$

$O(n)$ 이다.