알고리즘 실습 보고서

- MaxPQ -

전공 : 컴퓨터공학과

분반 : 05반

학번 : 201502085

이름 : 이규봉

1. 실행 환경

과제의 코드를 테스트한 환경은 아래와 같습니다. 따로 사용한 라이브러리는 없습니다.

- Windows 10
- Pycharm 2019.02
- Python 3.7.4

2. 과제 설명

- 우선순위 큐 구현 및 관련 CUI 프로그램 작성

3. 문제 해결 방법

- 문제를 해결하기 위해 자신이 사용한 방법, 아이디어에 대한 설명
  (구현한 코드에 대한 설명 등)

'함수명은 자유'로 되어 있어, 임의로 바꿔 구현했습니다. 바꾼 함수명과 매칭되는 (ppt내에서의 함수 이름) 이름들을 함께 기재했습니다.

우선순위 큐를 클래스로 구현했습니다. 우선 순위 큐 내부엔 Node 클래스가 정의되어 있으며, Node 클래스는 우선순위 값과 String 타입의 item을 갖고 있습니다. 이 클래스에 정의되어 있는 public 메서드는, 가장 높은 우선순위를 갖는 항목을 제거하는 dequeue, 특정 item에 해당하는 항목을 제거하는 delete, 우선순위 큐에 새 Node를 삽입하는 enqueue, 특정 item의 우선순위를 증가시키는 increase_priority, 우선순위 큐가 비어 있는지 확인하는 is_empty, 우선순위 큐의 길이를 확인하는 length, 우선순위 큐 내의 배열의 값들을 출력하는 print, 우선순위 큐 내 가장 큰 우선순위 값을 갖는 항목의 item 값을 반환하는 retrieve가 있습니다.

enqueue는 (ppt에 나온 메서드 중 insert에 해당하는 메서드)로, 우선순위 큐가 내부적으로 갖고 있는 배열에 새 Node를 만들어 append 합니다. 그 후 반복문에서 max_heapify를 호출해 다시 Max Heap의 invariant를 보장할 수 있게 만듭니다.

delete는 배열을 순회하며, 인자로 주어진 item과 같은 항목을 갖는 Node가 있는지 검사합니다. 있다면 그 항목 이후의 항목들을 전부 앞으로 옮겨 씁니다. 그 후 self.__arr에서 pop해 (중복된) 마지막 원소를 제거하고 count를 1 감소시킨 후 반복문에서 max_heapify를 호출해 다시 Max Heap의 invariant를 보장할 수 있게 만듭니다.

increase_priority (ppt의 increase_key와 같은 역할을 하는 메서드) 도 비슷하게 동작합니다. 배열을 순회하며 item과 같은 값을 갖는 Node를 찾고, 찾은 Node의 우선순위를 증가시킨 후, 반복문에서 max_heapify를 호출해 다시 Max Heap의 invariant를 보장할 수 있게 만듭니다.

dequeue (ppt의 extract_max와 동일한 역할을 하는 메서드) 는 우선순위 큐에서 가장 높은 우선순위를 갖는 항목을 제거해야 하는데, 이는 항상 Max Heap의 루트 노드입니다. 따라서,

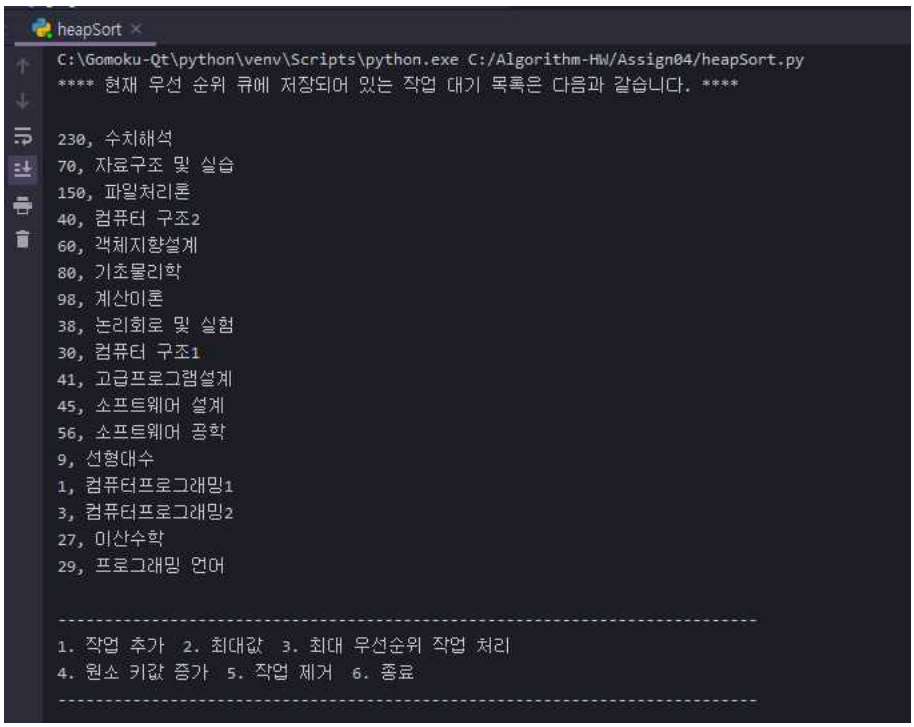
이 노드는 내부 배열의 0번째 인덱스에 위치합니다. 따라서 0 번째 원소를 delete 하고 반환하는 것으로 구현할 수 있습니다.

그리고 private 메서드로는, Max 힙이 아닌 힙을 Max 힙으로 만들기 위한 __max_heapify, 배열을 이용하여 우선순위 큐를 생성할 수 있게 해 주는 __init__, dequeue에서 가장 높은 우선순위의 노드를 제거하고 난 후에도 Max Heap을 유지할 수 있게 해 주는 재귀 함수인 __deque_recursive, 부모 노드의 인덱스를 반환하는 __get_parent_index가 있습니다.

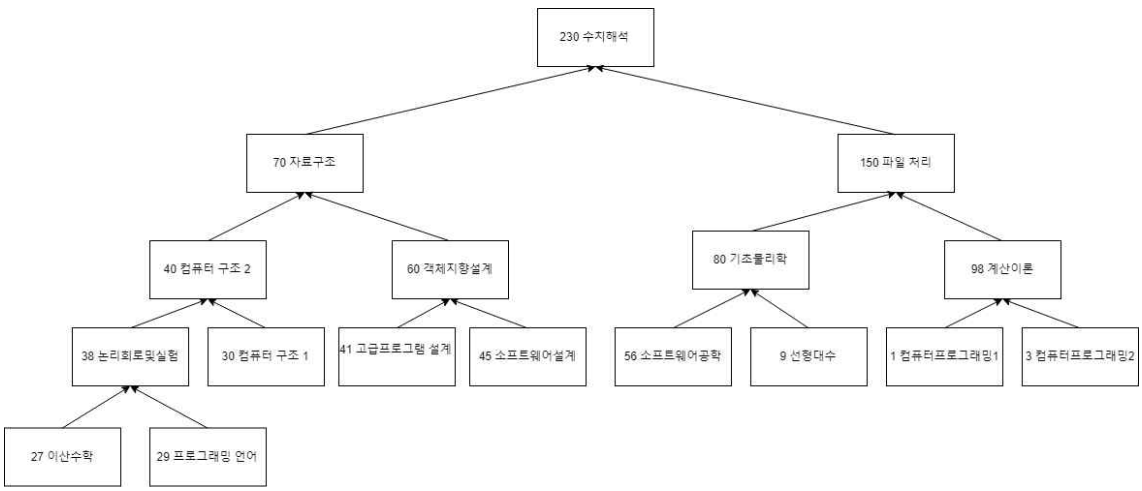
main 함수에선 무한 루프를 돌며, 1 ~ 6번 중의 값을 받아 각각의 경우에 해당하는 Priority_Queue의 public 메서드를 적절하게 호출합니다.

4. 결과 화면

1 – 기본 화면



위와 같은 Max Heap을 그림으로 나타내면 아래와 같습니다.
모든 노드들이 자신의 부모 노드보다 작다는 성질을 만족하므로, Max Heap의 정의에 만족됩니다.

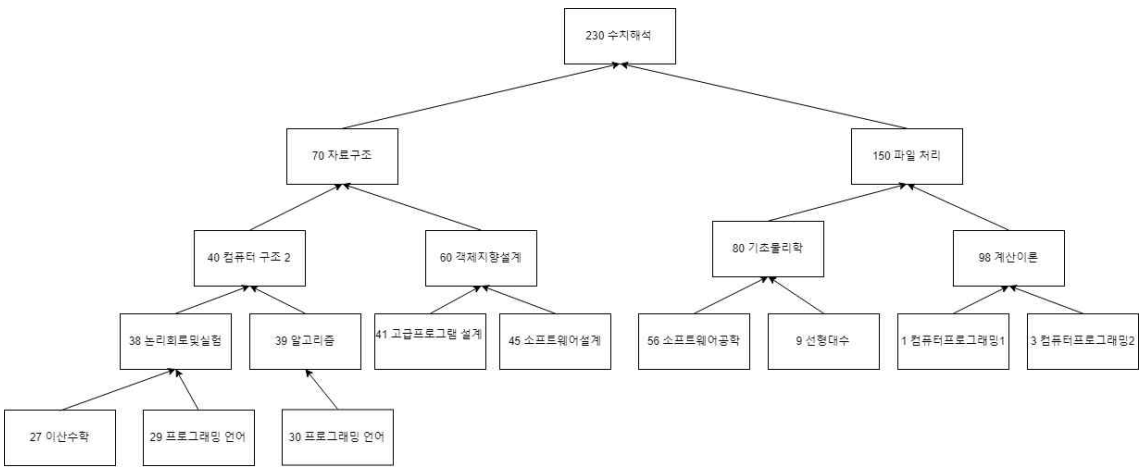


2 – 원소 삽입

```
--------------------------------------------------------------------------------
1
추가할 작업의 우선순위를 입력해주세요
39
추가할 작업의 이름을 입력해주세요
알고리즘
--------------------------------------------------------------------------------
**** 현재 우선 순위 큐에 저장되어 있는 작업 대기 목록은 다음과 같습니다. ****

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

39라는 우선순위와 함께 '알고리즘' 노드를 대입한 후의 우선 순위 큐의 대기 목록입니다.

우선순위 큐의 내부 Max Heap은 아래와 같이 변경됩니다.



여전히 Max Heap의 성질을 만족하는 것을 알 수 있습니다.

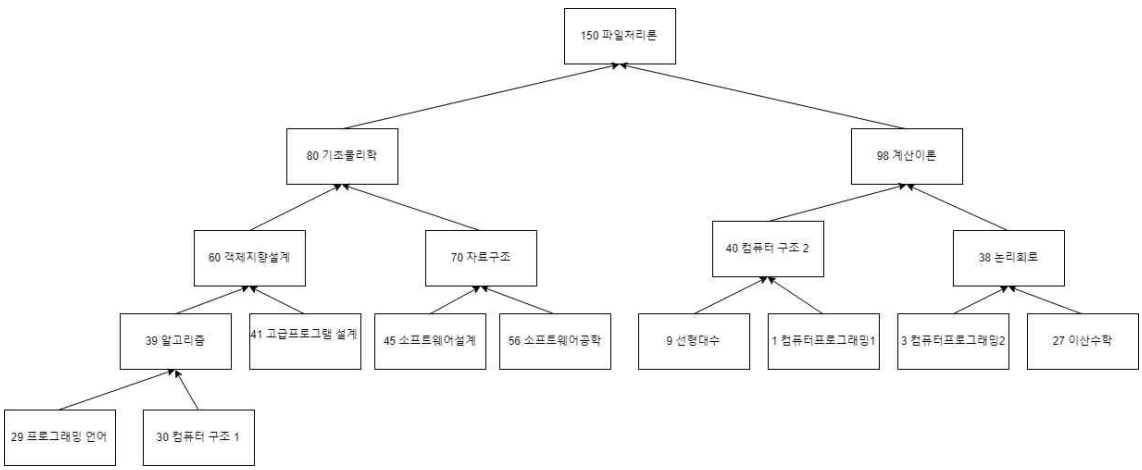
3 – 최대값 출력

```
--------------------------------------------------------------------------------
2
가장 높은 우선순위를 갖는 항목은 수치해석 입니다
--------------------------------------------------------------------------------
**** 현재 우선 순위 큐에 저장되어 있는 작업 대기 목록은 다음과 같습니다. ****

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

4 – 가장 높은 우선순위 작업 처리

```
--------------------------------------------------------------------------------
1. 작업 추가  2. 최대값  3. 최대 우선순위 작업 처리
4. 원소 키값 증가  5. 작업 제거  6. 종료
--------------------------------------------------------------------------------
3
가장 높은 우선순위의 작업을 처리했습니다
--------------------------------------------------------------------------------
**** 현재 우선 순위 큐에 저장되어 있는 작업 대기 목록은 다음과 같습니다. ****

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

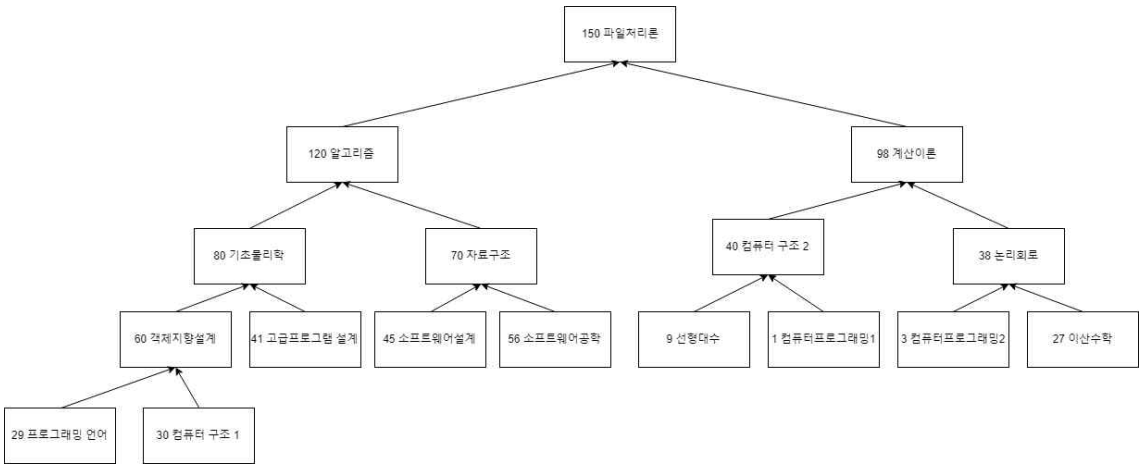
가장 높은 우선순위를 갖는 항목은 수치해석이므로, 수치해석 노드를 제거합니다.
우선순위 큐의 내부 Max Heap은 아래와 같이 변경됩니다.



5 – 키 값 증가

```
--------------------------------------------------------------------------------
4
증가시킬 항목의 이름을 입력해주세요
알고리즘
증가시킬 키 값을 입력해주세요 (감소는 불가)
120
--------------------------------------------------------------------------------
**** 현재 우선 순위 큐에 저장되어 있는 작업 대기 목록은 다음과 같습니다. ****

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

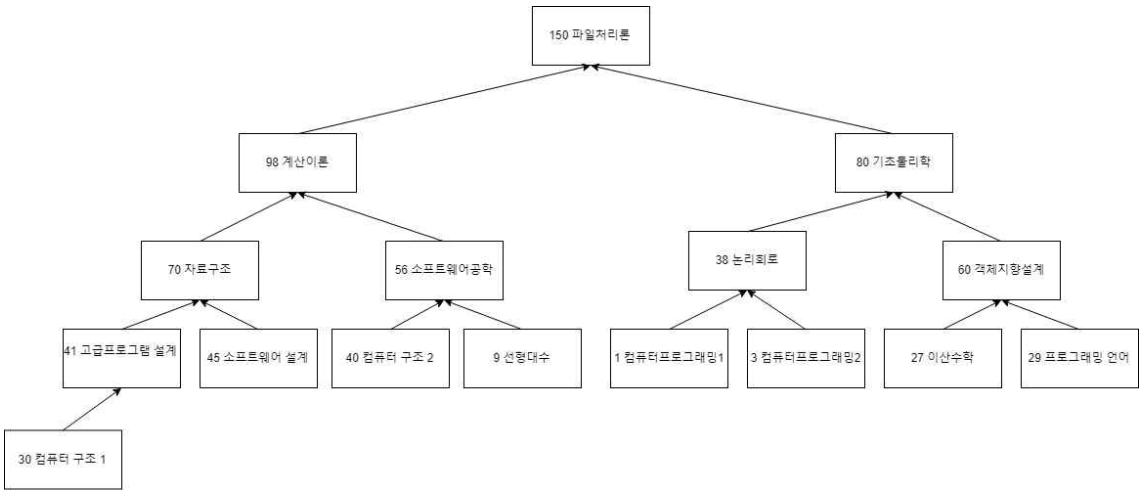
우선순위 큐의 내부 Max Heap은 아래와 같이 변경됩니다.



6 – 항목 제거

```
--------------------------------------------------------------------------------
1. 작업 추가  2. 최대값  3. 최대 우선순위 작업 처리
4. 원소 키값 증가  5. 작업 제거  6. 종료
--------------------------------------------------------------------------------
5
제거할 항목의 이름을 입력하세요
알고리즘
--------------------------------------------------------------------------------
**** 현재 우선 순위 큐에 저장되어 있는 작업 대기 목록은 다음과 같습니다. ****

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

우선순위 큐의 내부 Max Heap은 아래와 같이 변경됩니다.



이로써, 우선순위 큐에 1 ~ 6번 까지의 작업을 거쳐도 Max Heap의 invariant를 깨뜨리지 않는다는 것을 확인했습니다.

7 – 프로그램 종료

5. 느낀점 및 고찰

자료구조 시간에 배웠던 우선순위 큐를 다시 파이썬으로 구현해보며 이것저것 배울 수 있었습니다.

과제 수행에 걸린 시간은, Priority Queue 클래스 구현 및 나머지 코드 작성에 3 ~ 4시간 정도, 보고서 작성에 2 ~ 3시간 정도 소요되었습니다.