알고리즘

- HW02 : QuickSort,PriorityQueue-

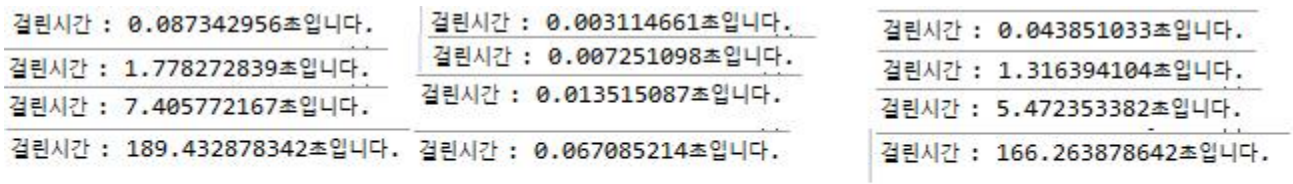
제 출 일	2017년 09월 28일
분 반	00반
담당교수	공은배
학 과	컴퓨터공학과
학 번	201302423
이 름	신종욱

1. 해결 방법

QuickSort의 경우에는 앞서 작성한 소트프로그램에서 소트부분만 수정하였다.
QuickSort 구현은 ppt에 나와있는 2번째 교수님이 가르쳐준 방법으로 구현하였다.
나와있는대로 그대로 알고리즘을 작성하였더니 바로 구현되었다.
Data_heap이 euc-kr코딩이라서 처음에 깨져서 이클립스 코딩을 바꿔주었다.
작년에 자료구조를 배우면서 max_heap을 구현한적이 있어서 이번과제에 알맞게 바꿔주었다.
단 없는 특정 데이터를 제거하거나 값을 바꿔주는거에대해선 없어서 추가 구현하였다.
그 과정에서 increase_key를 구현하는데 오래걸렸다 맥스힙의 중간에 큰값이 들어오면 root까지 올라가면서 heapify하는 형식으로 구현하였다.
delete는 삭제하고 리프노드를 올려서 heapify하는 형태로 구현하였다.
혹시 increase_key에 우선순위가 더 낮은거가 들어올경우에도 제대로 heapify하도록 if문을 추가하여서 구현했다.

2. 실행 결과

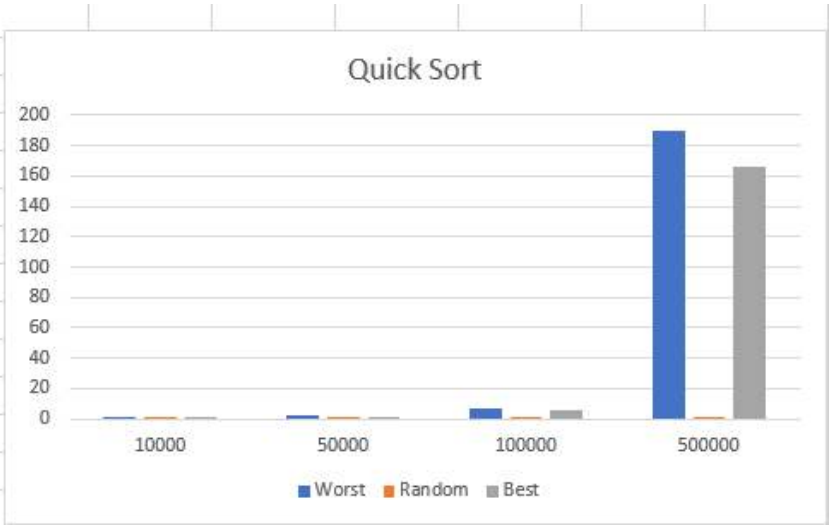
ㄱ.QuickSort



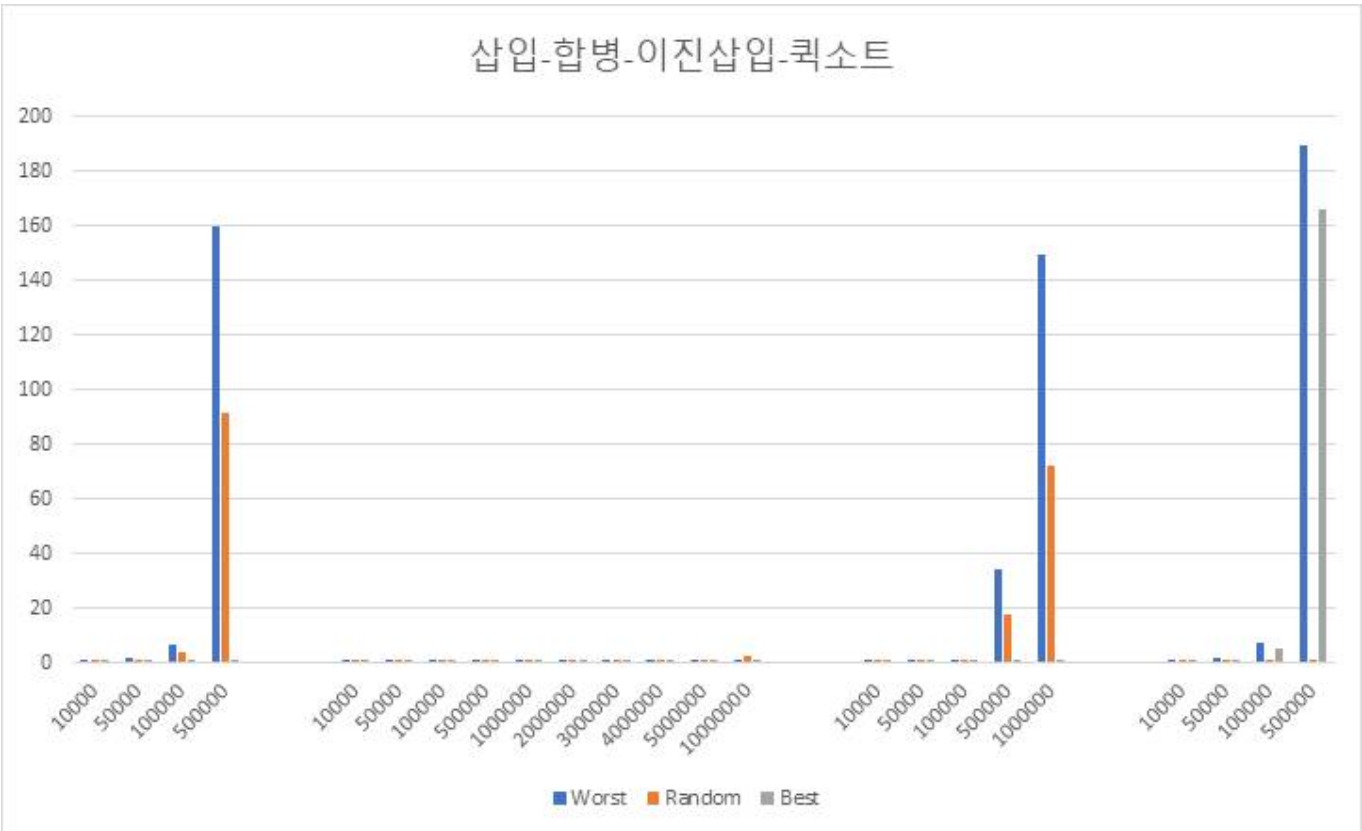
Worst Random Best

그래프분석

	Worst	Random	Best
10000	0.087343	0.003115	0.043851
50000	1.778273	0.007251	1.316394
100000	7.405772	0.013515	5.472353
500000	189.4329	0.067085	166.2639



QuickSort는 최악은 n^2인데 10만에서 50만으로 증가할 때 7초에서 189초로 증가한걸 보면 25배가량 증가한걸 볼 수 있다.Best인경우에도 quicksort입장에선 worst라서 상승률이 크다. 평균은 Nlogn복잡도인데 10만에서 50만으로 증가할 때 대략 5배증가한걸로 봐서 구현이 잘되었다.



다함께 볼 경우 퀵소트는 Worst의 경우에는 삽입정렬과 같은 복잡도를 가지게되어서 N이 커질수록 작업량이 매우크게 상승한다 제일 성능상 좋은 것은 합병정렬이지만 저장공간이 더 필요하다는 점을 생각해야한다. 랜덤일 경우 즉 평균일때는 퀵소트도 매우우수하기 때문에 저장공간이 한정적일땐 퀵소트도 좋을거 같다.

ㄴ.PriorityQueue

```
PriorityQueue [Java Application] C:₩Program Files₩Java₩jre1.8.0_144₩bin₩javaw.exe (2017. 9. 27. 오후 8:12:23)
**현재 우선 순위 큐에 저장되어 있는 작업 대기 목록은 다음과 같습니다.**
1- 230  수치해석
2 -  70   자료구조 및 실습
3 -  150   파일처리론
4 -  40   컴퓨터 구조2
5 -  60   객체지향설계
6 -  80   기초물리학
7 -  98   계산이론
8 -  38   논리회로 및 실험
9 -  30   컴퓨터 구조1
10 -  41   고급프로그램설계
11 -  45   소프트웨어 설계
12 -  56   소프트웨어 공학
13 -  9   선형대수
14 -  1   컴퓨터프로그래밍1
15 -  3   컴퓨터프로그래밍2
16 -  27   이산수학
17 -  29   프로그래밍 언어
---------------------------------------------------
1.작업 추가 2.최대값 3.최대 우선순위 작업 처리
4.원소 키값 증가 5.작업제거 6.종료
---------------------------------------------------
1
추가할 과목명을 입력해주세요알고리즘
우선순위를 입력하여주세요95
```

1

```
**추가후 현재 우선 순위 큐에 저장되어 있는 작업 대기 목록은 다음과 같습니다.**
1- 230  수치해석
2 -  95  알고리즘
3 -  150   파일처리론
4 -  70   자료구조 및 실습
5 -  60   객체지향설계
6 -  80   기초물리학
7 -  98   계산이론
8 -  38   논리회로 및 실험
9 -  40   컴퓨터 구조2
10 -  41   고급프로그램설계
11 -  45   소프트웨어 설계
12 -  56   소프트웨어 공학
13 -  9   선형대수
14 -  1   컴퓨터프로그래밍1
15 -  3   컴퓨터프로그래밍2
16 -  27   이산수학
17 -  29   프로그래밍 언어
18 -  30   컴퓨터 구조1
---------------------------------------------------
1.작업 추가 2.최대값 3.최대 우선순위 작업 처리
4.원소 키값 증가 5.작업제거 6.종료
---------------------------------------------------
4
몇번째 과목의 우선순위를 증가시키겠습니까??8
몇으로 변경하시겠습니까?200
```

2

```
PriorityQueue [Java Application] C:₩Program Files₩Java₩jre1.8.0_144₩bin₩ja

**증가시킨후 현재 우선 순위 큐에 저장되어 있는 작업 대기 목록은 다음과 같습니다.**
1- 230  수치해석
2 -  200   논리회로 및 실험
3 -  150   파일처리론
4 -  95  알고리즘
5 -  60   객체지향설계
6 -  80   기초물리학
7 -  98   계산이론
8 -  70   자료구조 및 실습
9 -  40   컴퓨터 구조2
10 -  41   고급프로그램설계
11 -  45   소프트웨어 설계
12 -  56   소프트웨어 공학
13 -  9   선형대수
14 -  1   컴퓨터프로그래밍1
15 -  3   컴퓨터프로그래밍2
16 -  27   이산수학
17 -  29   프로그래밍 언어
18 -  30   컴퓨터 구조1
---------------------------------------------------
1.작업 추가 2.최대값 3.최대 우선순위 작업 처리
4.원소 키값 증가 5.작업제거 6.종료
---------------------------------------------------
3
```

3

```
**처리를 완료하였습니다.**
1- 200  논리회로 및 실험
2 -  95  알고리즘
3 -  150   파일처리론
4 -  70   자료구조 및 실습
5 -  60   객체지향설계
6 -  80   기초물리학
7 -  98   계산이론
8 -  30   컴퓨터 구조1
9 -  40   컴퓨터 구조2
10 -  41   고급프로그램설계
11 -  45   소프트웨어 설계
12 -  56   소프트웨어 공학
13 -  9   선형대수
14 -  1   컴퓨터프로그래밍1
15 -  3   컴퓨터프로그래밍2
16 -  27   이산수학
17 -  29   프로그래밍 언어
---------------------------------------------------
1.작업 추가 2.최대값 3.최대 우선순위 작업 처리
4.원소 키값 증가 5.작업제거 6.종료
---------------------------------------------------
5
삭제하실 큐는 몇 번째 입니까?14
```

4

```
**삭제후 현재 우선 순위 큐에 저장되어 있는 작업 대기 목록은 다음과 같습니다.**
1- 200  논리회로 및 실험
2 -  95  알고리즘
3 -  150   파일처리론
4 -  70   자료구조 및 실습
5 -  60   객체지향설계
6 -  80   기초물리학
7 -  98   계산이론
8 -  30   컴퓨터 구조1
9 -  40   컴퓨터 구조2
10 -  41   고급프로그램설계
11 -  45   소프트웨어 설계
12 -  56   소프트웨어 공학
13 -  9   선형대수
14 -  29   프로그래밍 언어
15 -  3   컴퓨터프로그래밍2
16 -  27   이산수학
-----------------------------------------------------
1.작업 추가 2.최대값 3.최대 우선순위 작업 처리
4.원소 키값 증가 5.작업제거 6.종료
-----------------------------------------------------
2
우선순위가 가장높은 강의는  논리회로 및 실험입니다.
```

추가-증가-최대우선작업처리-삭제-최대값 출력순으로 실행해보았다.

4. 느낀 점

Worst와 Best의 시간이 오래걸려서 인터넷에 찾아보니 QuickSort의 경우 최악의 경우에는 N^2의 복잡도를 가진다고 하는거보니 바르게 구현한거같다 퀵소트 성능이 좋다고만 들어서 Worst일 경우도 좋은지 알았는데 이번 구현으로 아니란걸 알게되었다.
partition의 방법도 다양하다는걸 알게되었다.
max힙에서 선호도를 증가하였을 때 해당부분만 heapify하면 될줄알았는데
반복해서 root까지 해줘야하는걸 배우게되었다.
자료구조가 어떻게 생긴줄 아는게 모든 알고리즘을 이해하는거와 차이가 있다는걸 알았다.