

Homework #6

2020.10.12.

다음과 같은 자료구조를 사용한 탐색 알고리즘을 파이썬으로 구현한 다음 성능을 측정한다.

1. 이진 탐색 트리
2. 레드 블랙 트리
3. AVL 트리

※ 주의 사항

1. 보고서에는 다음과 같은 사항이 포함되어 있어야 함.
 - 실행 시간 측정 결과
 - 엑셀 등을 사용하여 측정 결과를 표시한 그래프
2. 3개 이상의 충분히 큰 N 값을 정하여 N 값의 변화에 따른 실행시간의 변화를 측정함.
3. 난수값을 가진 배열 외에도 정렬된 배열이라든지 큰 키와 작은 키가 번갈아 가면서 나오는 배열 등을 사용하여 최악의 경우에 대한 성능을 측정해 봄.
4. 프로그램의 실행 시간을 측정하기 전에 다음 예와 같이 레드-블랙 트리와 AVL 트리가 정확하게 구축되었는지 확인하는 함수를 추가함.

[프로그램 3.4] 레드-블랙 트리 탐색 프로그램에서 레드-블랙 트리가 정확하게 구축되었는지 확인하기 위해 Dict 클래스 아래에 check() 함수를 추가하여 다음과 같이 트리의 구조를 출력함

=====

[2, 1, 8, 9, 7, 3, 6, 4, 5]

key : 1 , parents: 2 , color : black

key : 2 , parents: 6 , color : black

key : 3 , parents: 4 , color : red

key : 4 , parents: 2 , color : black

key : 5 , parents: 4 , color : red

key : 6 , parents: 6 , color : black

key : 7 , parents: 8 , color : black

key : 8 , parents: 6 , color : black

key : 9 , parents: 8 , color : black

레드 블랙 트리 탐색의 실행 시간 (N = 9) : 0.000

탐색 완료

=====

[프로그램 3.5] AVL 트리 탐색 프로그램에서 AVL 트리가 정확하게 구축되었는지 확인하기 위해 Dict 클래스 아래에 check() 함수를 추가하여 다음과 같이 트리의 구조를 출력함

```
=====
[2, 1, 8, 9, 7, 3, 6, 4, 5]
key : 1 , parents: 2
key : 2 , parents: 4
key : 3 , parents: 2
key : 4 , parents: 7
key : 5 , parents: 6
key : 6 , parents: 4
key : 7 , parents: 7
key : 8 , parents: 7
key : 9 , parents: 8
AVL 트리 탐색의 실행 시간 (N = 9) : 0.000
탐색 완료
=====
```