

극단적인 케이스 (균형이 깨진 [편향 이진 트리])에서의 실행시간 비교 (0~4509)	입력을 test.txt 로 받은 상태에서의 실행시간 비교
<p>인터넷에서 찾은 RB tree</p> <ul style="list-style-type: none"> : 2.378 : 2.444 : 2.490 : 2.479 : 2.413 <p>=> 2.4408</p>	<p>인터넷에서 찾은 RB tree</p> <ul style="list-style-type: none"> : 1.596 : 1.522 : 1.592 : 1.593 : 1.599 <p>=> 1.5804</p>
<p>인터넷에서 찾은 AVL tree</p> <ul style="list-style-type: none"> : 2.332 : 2.309 : 2.319 : 2.329 : 2.315 <p>=> 2.3208</p>	<p>인터넷에서 찾은 AVL tree</p> <ul style="list-style-type: none"> : 1.618 : 1.717 : 1.562 : 1.615 : 1.585 <p>=> 1.6194</p>
<p>인터넷에서 찾은 binary_search tree</p> <ul style="list-style-type: none"> : 2.619 : 2.552 : 2.542 : 2.560 : 2.524 <p>=> 2.5594</p>	<p>인터넷에서 찾은 binary_search tree</p> <ul style="list-style-type: none"> : 1.581 : 1.544 : 1.538 : 1.481 : 1.575 <p>=> 1.5438</p>
<p>직접 구현한 Rb_tree</p> <ul style="list-style-type: none"> : 2.348 : 2.191 : 2.375 : 2.310 : 2.304 <p>=> 2.3056</p>	<p>직접 구현한 Rb_tree</p> <ul style="list-style-type: none"> : 1.535 : 1.570 : 1.543 : 1.545 : 1.517 <p>=> 1.5420</p>

분석

최대한 동일한 환경에서 실행시간이 비교될 수 있게 하였으며, 크게 차이는 나지 않지만 전반적으로 직접 짠 코드의 실행시간이 적게나온 것을 확인할 수 있다.

분석을 통해 놀랐던 사실은 케이스가 크게 극단적이지 않은 테스트 케이스가 들어올 경우, binary search tree와 RB tree의 실행시간은 크게 차이나지 않는다는 점이었다.

이는 직접 짠 코드가 최적화된 코드가 아니거나, 인터넷에서 찾은 binary search tree가 최적화 된 코드이거나의 문제일 수도 있고, RB tree는 들어온 데이터에 대해 '재정렬'하는 시간이 필요하므로 극단적으로 치우치지 않은 이상 재정렬을 할 필요없는 binary search tree랑의 실행시간 차이가 없는 것일수도 있다.