삭제하려면 부모 포인터를 알고있어야함. 그래서 보통은 재귀로

자식이 아예 없거나, 1개라면 문제될게 없음. 그냥 연결하면 되니까. 근데 문제는 자식이 2명 이상일 때.

값만 복사할 것인가

노드를 옮길것인가.

근데 그냥 값만 복사해라. 객체일경우 수많은 대입연산자가 불릴텐데 그건 그렇게 쓴사람이 잘못

그러니까 그냥 값을 대입하고, 원래 애를 지워버려라.

누구를 대체해야 하냐면, 왼쪽에서 가장 큰 데이터, 오른쪽에서는 가장 작은 데이터. 40을 삭제한다면 36또는 41. left의 끝단, right의 끝단. 양쪽에서 제일 리프노드.

right에있는 제일 작은애로 대체시키겠다면 right로 한번 가고 계속 left로 타고 가면서 null이 나올때까지 찾는다. 거기서 right가 달려있어도 내 위랑 연결해주고 left 제일끝은 위로 올라감.

여기서 분기를 또 나누어야하느가? 그럴 필요가 없는데. 그냥 부모의 left를 올라갈 애의 right를 대입하고 퉁친다. null이든 데이터가 있든 그냥 대입.

레드블랙트리에서도 삭제규칙은 동일. 밸런스잡는 코드가 추가될 뿐이다

레드블랙트리는 색깔이 규칙을갖고있음. 레드와 레드를 연속으로 오지 못하게 하는 규칙을 갖고 있음. 또 모든 경로를 대상으로 블랙의 개수는 동일하다.

여기서 최악은 한쪽이 블랙으로만 이루어진 상황. 다른한편은 레드와 블랙이 교차되며 있는상황.

삽입하기위해 끝단을 찾는데, nil의 위치에 새 노드를 넣는다.

밸런스를 잡을땐 nil노드를 진짜 있는 블랙 노드로 인지하고 구조를 잡아야한다.

nil을 모든 끝단에 넣어야하는데 그러기 번거로우니 하나로 퉁치자. 모든 끝단의 left right의 주소는 nil의 주소를 넣자.

삽입을 하면 삽입된 노드가 밸런스의 시작. 삭제를 하면 삭제된 노드의 자리를 대체한 녀석이 밸런스의 시작. 근데 만약 자식이 없는 마지막 leaf노드를 삭제하면 그 밑에있는 nil을 기준으로 밸런싱.

근데 그 럼 닐은 부모가 없었는데. 닐은 모든 노드에 연결되어있음. 모두를 바라보고있음. 공용의 nil로 밸런싱하기가 이상하다. 누군가가 삭제되고 그 위치에 nil이 들어오는 상황이라면

공용으로 쓰는 nil노드가 방금삭제한 애의 부모가 되어있을것. 그럼 nil을 기준으로 밸런스 잡아도 된다는것. 그래도 되나? 그래도 된다. nil의 부모를 쓰는건 삭제하는경우.

평소엔 위에서 아래로 내려오는 방향이라 괜찮다. 다른애가 삭제되면 nil의 부모는 또 바뀔것.

트리의 회전이 필요하다

10

5 20

4 6

이렇게 있으면

5

4 10

6 20

이렇게 된다 우회전되면 .

그래서 사전재료로 회전함수가 필요하다.

새로 노드가 추가되는건 레드. 왜? 첫조건은 블랙 개수를 맞추는것. 그래서 블랙이 들어가면 무조건 틀어짐. 레드를 넣은다음에 레드레드 인 경우에만 밸런스를 맞추면 되니까.

새 부모의 노드가 레드라면 레드레드니까 조건을 타고 들어감.

할아버지를 기준으로 밸런스를 맞추는걸 목표로한다. 왜? 할아버지 기준으로 좌우 밸런스 맞추기 힘들면 할아버지 색을 변경시키며 위로 올라갈거임. 어려운 상황ㅇ느 내랑 부모가 레드레드라 블랙이 하나 추가되어야하는 상황

그래서 할아버지 기준 노드 아래에서 기존 블랙은 변화없이 좌우 블랙을 맞추고싶은것. 블랙을 하나 빼고싶다면 할아버지를 레드로 바꾸는것. 그럼 할아버지 부모랑 레드레드일수있는거 그래서 위로 쭉 올라가야함.

1. 나랑 부모랑 레드레드인 경우 . 할아버지 컬러는 블랙. 100% 그리고 내 형제는 블랙. 부모가 레드였기때문에. 근데 여기서 분기는 삼촌이 레드 블랙인경우

여기서 삼촌이 블랙 - 삼촌이 nill인 경우. (나 부모 할아버지 삼촌 - 레드 레드 블랙 블랙인데 그럼 개수가 안맞는상황. 그러면 삼촌은 nill경우) 그래서 블랙의 밸런스를 유지하면서 레드레드를 바꾸는 상황. 그래서 삼촌이 뭔지를 먼저 판단해야함.

여기서 삼촌이 레드 - 그럼 나 부모 할아버지 삼촌이 레드 레드 블랙 레드 인 경우 . 블랙 개수를 1개로 유지하면서 레드 레드인 경우를 없애려면? 부모와 삼촌을 블랙으로 바꾸고 할아버지를 레드로.

근데 그러면 할아버지의 부모가 레드이면 또 문제. 그래서 루트까지 올라가면서 해결해야한다. 그럼 이제 할아버지를 다시 newnode 처럼 잡아야한다. 왜? 아까 새로운 노드 들어오면서 레드라고 하고 작업한거나 마찬가지니까. 할아버지를 newnode 포인터로 대입하고 다시 이 작업 시작.

그럼 내가 잡은 newnode는 레드로 시작했고, 그 newnode의 부모가 black이면 끝난것. 근데 레드면 또 이 작업 반복.

삼촌이 블랙 - 나 부모 할아버지 삼촌 (레드 레드 블랙 블랙) 이 상황은 불규칙같지만 사실 블랙 대칭이 맞는것. 그럼 이상황은 회전해야한다. 부모를 할아버지 위치로 오른쪽으로 회전.

그럼 나 부모 할아버지 삼촌일때 (부모를 블랙으로, 할아버지를 레드로 만들면 대칭이 맞는다. 이렇게 내부로 끝낼수있음. )

근데 문제는 내가 레드인데 부모의 오른쪽에 들어간 경우. 이건 절대 해결을 못한다. 우회전 하면 삼촌 자식으로 들어가게됨. 그래서 부모를 밑으로 밀고 내가 부모자리로 들어가야한다.

그리고 그 상황에서 newnode는 더이상 내가 아니라 부모가 됨. 위에 문제랑 똑같은 상황이 되었으니까.