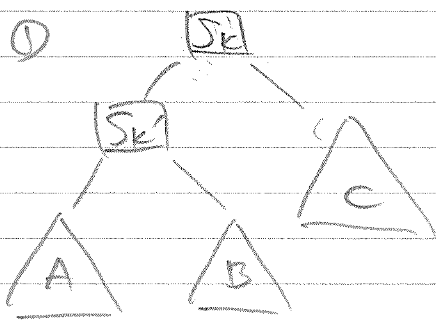
二分木の平衡化

東大理II 2年 栗本英理子

先週の議論で扱った二分木の平衡化について、考えたことをまとめました。

$\boxed{\text{定義}}$ ノードの集合をS、各要素をSkとおく
ノードSkの深さを根からの距離(枝の数)で定め、
これをhk、この集合をHとおく
Hの最大値をhmax、最小値をhminとおく
「Sk以下の木」をSk自身、Skの子とその子たちからなる木とする
操作 回転 について

①

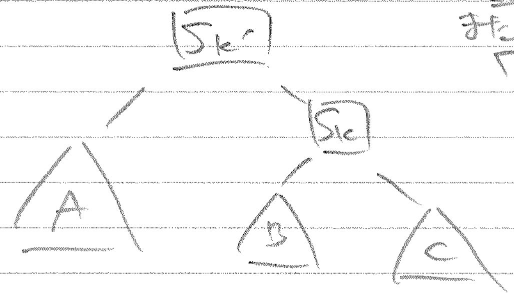


任意の葉でないSk以下の木について
左図のように書ける(A,B,Cが空のこともある)
ここでSk'以下の方がCより
深いときを考える
(反対でも一般性を失わない)

ここで、あるノード以下の木のノードでもっとも
深いノードの深さが大きいことを
「重い」とし、小さいことを「軽い」とする
また子がないときはもっとも軽い」

⇓

②



「Skに関して回転する」とは、
Skの子2つのうちより重い方のノードを
Sk'とし、左図のようにSk'を
根とした木に組みかえることである
(重さが同じ場合には回転の方向は
指定する)

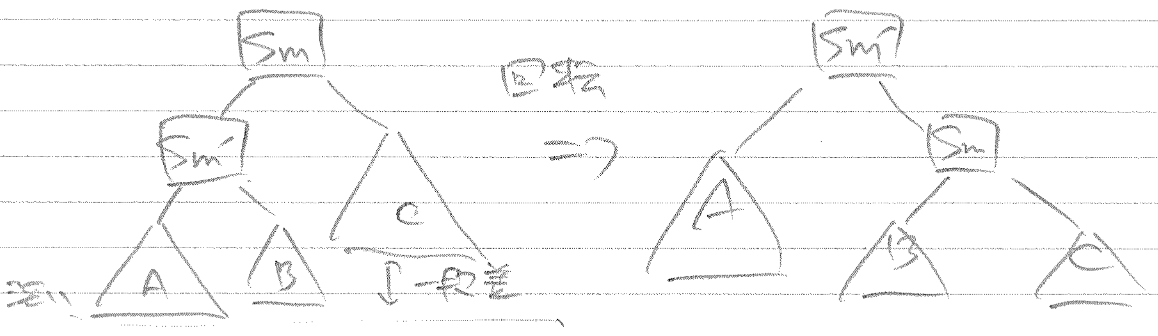
回転の操作をしても二分木がこわれないことを示す
ここでは左の方が重いときを考える(逆も同様)
二分木の定義より Sk'、Aの中のノード、Bの中のノードはいずれも
Skより小さく、Cの中のノードはSkより大きい
すると Sk'を根にしたときも上の図②でもこの関係が保たれる。

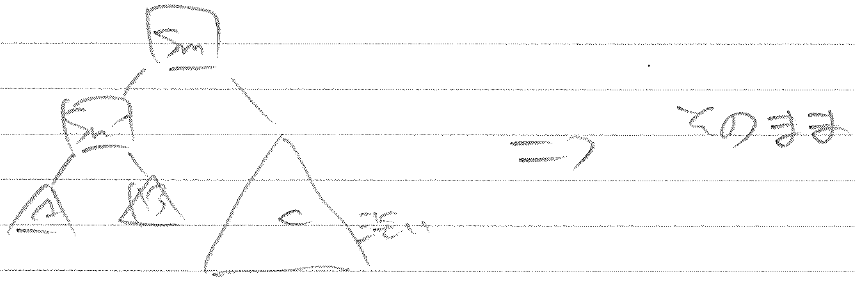
さらに「回転」の特徴をまとめる
計算量は Sk'の子1つとSkの子2つを組みかえるだけなので O(1)
重い方のノードが軽い方のノードより小さい場合、回転をすると、
重い方のノードの子2つのうち小さい方は1つ分軽くなり、大きい方は
不変、軽い方のノードは重くなる
木が反転しても然り (図②ではAが軽く、Bは同じ、Cは重く)

ルール　$h_{max} - h_{min} \leq 1$ のときは許容し、平衡化されているとする.
$h_{max} - h_{min} \geq 2$ となったときに平衡化を行う
この場合 $h_{max} - h_{min} > 2$ となることはなく $h_{max} - h_{min} = 2$ になる
このとき $h_{max} = h_k$ をみたすノードS'はただ1つ存在する
(のちに証明する)
ゆえに S'を子でその子たちに含む任意のノード S_k について
重い方の子と軽い方の子が定義でき、回転が無条件に成立する

③ まず根のノードについて、重い方のノードS''に注目をうつす
このS''以下には必ずS'が含まれているので、重い軽いが定義できる
このとき重い方のノードの木の h_k の最大と軽い方のノードの h_k の
最小の差が2となった場合は、前と同じように重い方のノードS''に
注目をうつす、ということを以下同様につづける

もし差が1となった場合について
④ そのノード S_m が S_m の親にとって小さい方の子だった場合、
S_m の子で重い方が小さい方のときはそのまま、
重い方が大きい方のときは S_m について回転する.
大小逆もまた然り
以下にイメージをかく (S_m が大きい方のとき)





のちに証明　すると、S_m の親に注目をうつしたとき、必ず定義の図①に対して
AかBより一段分重くなる形になり、また③の操作より必ず S_k の子で
重い方軽い方が定義できる。これまた④の操作をする。
ここで S_m の親の S_k で回転できる これをどんどん続けて根まで回転させる 終了.

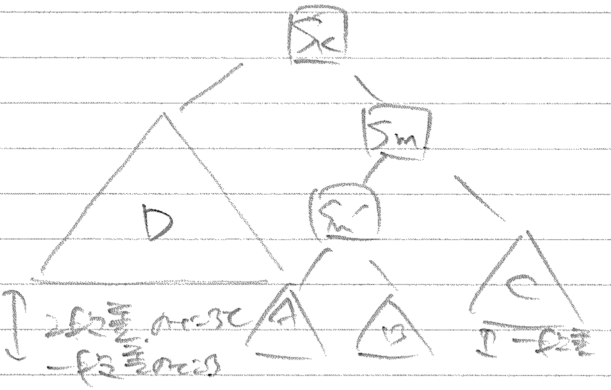
証明 補題1 $h_{max}-h_{min}\le 2$ を示す

最初は根のノードがあるだけなので $h_{max}=h_{min}=0$ で
$h_{max}-h_{min}=0\le 1$
以下、あるノードを足す直前に $h_{max}-h_{min}\le 1$ であるとすると、
次は1つノードを足すだけなので h_{max} は高々1しかふえず
h_{min} がへることはない
よって $h_{max}-h_{min}\le 2$ である
このとき平衡化を行うので、この平衡化が必ず高さの差を1
へらすので、$h_{max}-h_{min}\le 1$ となる
ゆえに数学的帰納法より $h_{max}-h_{min}\le 2$ で
平衡化を行うのは $h_{max}-h_{min}=2$ のときに限る

補題2 $h_{max}-h_{min}=2$ のとき $h_{max}=h_c$ をみたすノードはただ1つ

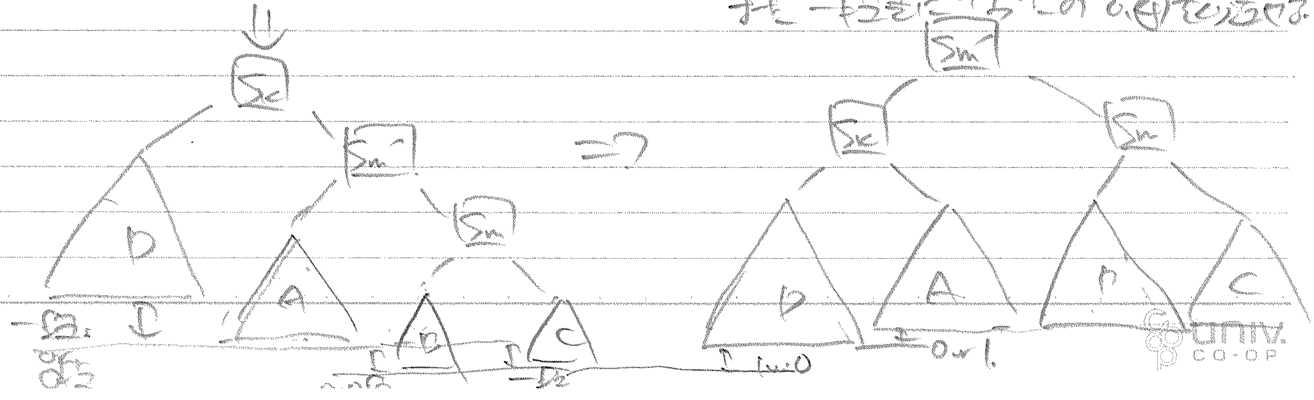
補題1より $h_{max}-h_{min}=2$ になる直前は $h_{max}-h_{min}=1$
すると、ここから h_{max} が1ふえると $h_{max}-h_{min}=2$ になる
ここで、足せるノードはただ1つなので、1ヶ所のみで h_c が
ふえることになるので、$h_{max}=h_c$ をみたすノードはただ1つであった

補題3 一段差で回転するといい感じになるやつ



③より A,BとDの差は2段、
A,BとCの差は1段

ここでこのような関係のときは、
S_m で回転することで
左下図のようになることが
保証される
すると S_c で回転して 下図のようになり
また一段差になったので 中を満たす



このルールで平衡化ができることを示す

「降りるパート」と「昇るパート」にわける

「降りるパート」ではルールにおける③をする
「昇るパート」ではルールにおける④のくりかえしをする

「降りるパート」では、補題1,2を用いることで、③をみたす操作と条件分岐ができると分かる。
ゆえに可能。

「昇るパート」では、補題3より、「降りるパート」でたどったノードをすべてたどっていくことで、であうのノードで④を行ったあと必ず深さのmaxとminの差が1となり、またそこの上位の深さより同じ、または軽くなることが保証されている。
これを根まで行うと、根以下の木について重い方と深い方の差が1以下になるということであり、このとき、
$h_{max} - h_{min} \leqq 1$ を満たす
ゆえに平衡化が可能になっている。

この平衡化が $O(\log N)$ でできることを示す

まず最初に、すべてのノードについて重さを保存しておく。
これは新たな要素を追加する際、その前の情報を保存しておいたものを用いればよいので $O(1)$ で分かる。

「降りるパート」について
③ 1回は、重さの比較をするだけなので $O(1)$
また、高さ 根から葉までの一本道をすべてたどるのみなので
$O(\log N)$ 回 ③をおこなうので
計算量は $O(\log N)$

「昇るパート」について
④ 1回は比較したあと回転するのみで、回転は $O(1)$ でできることより $O(1)$。回転したときは重さをかきかえるがこれも $O(1)$。
またこれを高々 $O(\log N)$ 回しかしないので、
計算量は $O(\log N)$

ゆえに2パートあわせて計算量は $O(\log N)$