

RB\_INSERT\_FIXUP (T, z)

while (z.p is **RED**) and (z ≠ T.root)

if z.p is a **LEFT** child

y is an **RIGHT** uncle

if y is **RED**

<case 1>

else

if z is a **RIGHT** child

<case 2> then continue to <case 3>

<case 3>

else // z.p is a **RIGHT** child

// same as above but **RIGHT** ⬄ **LEFT**

y is an **LEFT** uncle

if y is **RED**

<case 1’>

else

if z is a **LEFT** child

<case 2’> then continue to <case 3’>

<case 3’>

T.root = **BLACK**

case 1, 1’:

z.p.color = **BLACK**

y.color = **BLACK**

z.p.p.color = **RED**

z = z.p.p

case 2:

z = z.p

LEFT\_ROTATE (T, z)

case 2’

case 3:

z.p.color = **BLACK**

z.p.p.color = **RED**

RIGHT\_ROTATION (T, z.p.p)

case 3’

T(n) = a.T(n/b) + f(n)

a≥1,b>1,f(n)>0,≠>0 as n->∞

Compare nlogba vs. f(n)

Case 1: nlogba > f(n)

T(n) = Ꝋ(nlogba)

Case 2: nlogba = f(n)

T(n) = Ꝋ(nlogba\**lg*n)

Case 3: nlogba < f(n)

T(n) = Ꝋ(f(n))

T(n) = 2T() + Ꝋ(n) = Ꝋ(n*lg*n)

Upper bound:

Guess T(n) = O(n*lg*n)

T(n) ≤ dn*lg*n (d: pos. const.)

T() ≤ d*lg*

Substitute:

T(n) ≤ 2d()*lg* + cn

= dn*lg*+cn

=dn(*lg*n – 1) + cn

≤ dn*lg*ln if (-dn+cn≤0)

=> c ≤ d

∴ T(n) = O(n*lg*n)

T(n) = 2T() + Ꝋ(n) = Ꝋ(n*lg*n)

Lower bound:

Guess T(n) = Ω(n*lg*n)

T(n) ≥ dn*lg*n (d: pos. const.)

T() ≥ d*lg*

Substitute:

T(n) ≥ 2d()*lg* + cn

= dn*lg*+cn

=dn(*lg*n – 1) + cn

≥ dn*lg*ln if (-dn+cn≥0)

=> c ≥ d

∴ T(n) = Ω(n*lg*n)

∴ T(n) = Ꝋ(n*lg*n)

T(n) = 8T() + Ꝋ(n2) = 8T() + cn2

Upper bound:

Guess: T(n) = O(n3)

T(n) ≤ dn3 (d > 0)

T() ≤ d()3

Substitute:

T(n) ≤ 8(dn3/8) +cn2 = dn3+cn2

≤ dn3 if cn2 ≤ 0 => no c and n exist

New guess (–a lower-order term)

T(n) ≤ dn3 – d’n2 (d, d’>0)

T() ≤ d()3 – d’()2

= dn3/8 – d’n2/4

Substitute: T(n) ≤ 8T() + cn2

≤ 8(dn3/8 – d’n2/4) + cn2

= dn3 – 2d’n2 + cn2

≤ dn3 – d’n2 if (–d’n2 + cn2) ≤ 0

=> c ≤ d’ ∴T(n) = O(n3)

T(n) = 8T() + Ꝋ(n2)

= 8T() + cn2

Lower bound:

Guess: T(n) = Ω(n3)

T(n) ≥ dn3 (d > 0)

T() ≥ d()3

Substitute:

T(n) ≥ 8(dn3/8) +cn2= dn3+cn2

≥ dn3 if cn2 ≥ 0

=> always happens

∴T(n) = Ω(n3)∴T(n) = Ꝋ(n3)

**T(n) = 4T()+n => O(n*lg*n)**

Running time of Quicksort

T(n) = Ꝋ(n)+T(q-1)+T(n-q)

= T(αn) + T((1-α)n) + Ꝋ(n)

Mergesort T(n) = 2T() + Ꝋ(n)

T(n)= T(n/2)+c

∴Ꝋ(*lg*n)

T(n)=4T(n/2)+cn

∴Ꝋ(n2-n)=Ꝋ(n2)

T(n)=T(αn)+T((1-α)n)+cn

∴O(n*lg*n)

Counting sort <6, 0, 2, 6, 0, 8>

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A= | 6 | 0 | 2 | 6 | 0 | 8 |  | C= | 2 | 0 | 1 | 0 | 0 | 0 | 2 | 0 | 1 |
| idx | 1 | 2 | 3 | 4 | 5 | 6 |  | idx | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| B= | - | - | - | - | - | 8 |  | C= | 2 | 2 | 3 | 3 | 3 | 3 | 5 | 5 | ***6-1*** |
|  | 1 | 2 | 3 | 4 | 5 | 6 |  |  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| B= | - | 0 | - | - | - | 8 |  | C= | ***2-1*** | 2 | 3 | 3 | 3 | 3 | 5 | 5 | 5 |
|  | 1 | 2 | 3 | 4 | 5 | 6 |  |  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| B= | - | 0 | - | - | 6 | 8 |  | C= | 1 | 2 | 3 | 3 | 3 | 3 | ***5-1*** | 5 | 5 |
|  | 1 | 2 | 3 | 4 | 5 | 6 |  |  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| B= | - | 0 | 2 | - | 6 | 8 |  | C= | 1 | 2 | ***3-1*** | 3 | 3 | 3 | 4 | 5 | 5 |
|  | 1 | 2 | 3 | 4 | 5 | 6 |  |  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| B= | 0 | 0 | 2 | - | 6 | 8 |  | C= | ***1-1*** | 2 | 2 | 3 | 3 | 3 | 4 | 5 | 5 |
|  | 1 | 2 | 3 | 4 | 5 | 6 |  |  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| B= | 0 | 0 | 2 | 6 | 6 | 8 |  | C= | 0 | 2 | 2 | 3 | 3 | 3 | ***4-1*** | 5 | 5 |
|  | 1 | 2 | 3 | 4 | 5 | 6 |  |  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|  |  |  |  |  |  |  | final | C= | 0 | 2 | 2 | 3 | 3 | 3 | 3 | 5 | 5 |
|  |  |  |  |  |  |  |  |  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |