Before Changes

update() - if item does not exists so you need to add it as a new record with grow()

search of(5n + 3) + grow of(3n + 8) + sort of(21n2 + 7n +2) + 9 == 21n2 + 7n + 20

dropping the constant and lower variables makes this a quadratic operation

T(n) = O(n2)

update() - if item does not exists so you need to add it as a new record, without grow()

// 12 static search (5n + 3) + sort of (21n2 + 7n + 2) + 9 //21n2 + 12n + 14 T(n) = O(n2)

update() - if item does exists and you are just modifying what is there

T(n) =(5n + 3) + 5 T(n) = O(n)

find() - if item is there

5 static search of (5n + 3) // 5n + 7

T(n) = O(n)

find() - if item is not there

3 static search of (5n + 3)//5n + 6

T(n) = O(n)

remove() - if item is there

(5n + 7) + search of (5n + 3) = 10n + 10

remove() - if item is not there

search of (5n + 3) + 4 = (5n + 7)

copy constructor

2n(5n + 8) + 12 = 10n2 + 16n + 5

assignment operator

+ remove for n times 2n(10n + 10) + update with search and sort ( 21n2 + 7n + 20) + 3n + 12 == 21n2 + 10n + 27

T(n) = O(n2)

Destructor

3n(5n +7) + 5 == 15n2 + 21n + 12

T(n) = O(n2)

After Changes

update() - if item does not exists so you need to add it as a new record with grow()

search of O(logn) + grow of(3n + 8) + sort of O(nlogn) comes out to a worst case of T(n) = O(n)

update() - if item does not exists so you need to add it as a new record, without grow()

search of O(logn) + sort of O(nlogn) comes out to a worst case of T(n) = O(nlogn)

update() - if item does exists and you are just modifying what is there

Search of O(logn) + 5 static operations comes to a worst case of T(n) = O(logn)

find() - if item is there

Search of O(logn) + 5 static operations comes to a worst case of T(n) = O(logn)

find() - if item is not there

Search of O(logn) + 4 static operations come to a worse case of T(n) = O(logn)

remove() - if item is there

Search of O(logn) + (5n + 7) has a worst case of T(n) = O(n)

remove() - if item is not there

Search of O(logn) + 4 constants is a worse case of T(n) = O(logn)

copy constructor

5 operations done in loop for size n and 5 constants == 5n + 5 worse case T(n) = O(n)

assignment operator

3n for size of array in first loop, 5n in second loop for size of new array with 12 constants 3n + 5c + 12 worse case T(n) = O(n)

Destructor

3n for size of array of elements 5 constants 3n+5 is worse case of T(n) = O(n)

New runtimes

