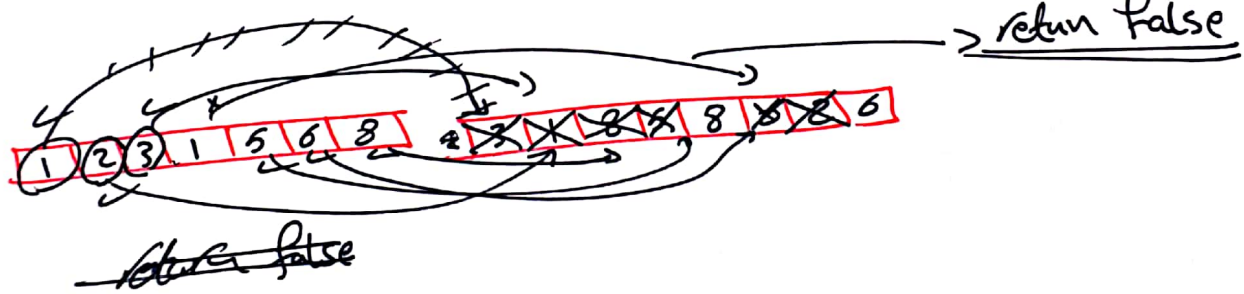
Assignment 6

C-413 Sequence A , Sequence B , size = n

Duplicates

Algorithm compareSameElements(A, B)

```
for i:=0 to A.size() do
    indexB := B.indexOf(A[i])
    if indexB = -1 then
        return false
    B.remove(indexB)
if ¬B.isEmpty() then
    return false
return true
```



~~return false~~

* $T(n) = 2T(n/2) + \log n \rightarrow \log_2 2 = 1$

case 1 :- $f(n) = O(n^{1-\epsilon}) \rightarrow O(n^0) \rightarrow O(1) \rightarrow$ X $\quad 0 < \epsilon \leq 1$

case 2 :- $f(n) = \Theta(n \log^k n) \rightarrow \Theta(n \log^2 n)$ ✓ $\quad k \geq 0 \rightarrow k = 1$

$T(n) = 8T(n/2) + n^2 \rightarrow \log_2 8 = 3$

case 1 :- $f(n) = O(n^{3-\epsilon}) \rightarrow O(n^2) \rightarrow$ ✓

$T(n) = 16T(n/2) + (n \log n)^4 \rightarrow \log_2 16 = 4$

case 1 :- $f(n) = O(n^{4-\epsilon}) \rightarrow O(n^3) \rightarrow$ X

case 2 :- $f(n) = \Theta(n^4 \log^k n) \rightarrow \Theta(n^2 \log^2 n)$ for $k \geq 1$ ✓

$T(n) = 7T(n/3) + n \rightarrow \log_3 7 = 1.77$

case 1 :- $f(n) = O(n^{1.77-1}) \rightarrow$ X

case 2 :- $f(n) = \Theta(n^{1.77} \log^k n) \rightarrow$ ✓

$T(n) = 9T(n/3) + n^3 \log n \rightarrow \log_3 9 = 2$

case 1 :- $f(n) = O(n^{2-1}) = O(n) \rightarrow$ X

case 2 : $f(n) = \Theta(n^2 \log^k n) \rightarrow$ ✓