3c) i) [0] n=0

resize → [1] n=1

resize → [1,1] n=2

[1,1,0] n=3

[1,1,1,1] n=4

resize → [1,1,1,1,0,0,0,0] n=5

⋮

[1,1,1...0,0,0] n=n

n levels

resizes cost $O(n) + O(n/2) + O(n/2^2) \dots$

$= O(n)$

total cost: $O(n) + O(n) = O(n)$

[1,1,1,...0,0,0] n=n

⋮

resize → [1,1,1,1] n=4

O(1) – [1,1,1,0] n=3

O(1) – [1,1,0,0] n=2

resize → [1,0] n=1

[0] n=0

$O(n)$

resizes cost $O(n) + O(n/2) + O(n/2^2) \dots$

$= O(n)$

total $= O(n) + O(n) + O(n)$

$= \boxed{O(n)}$

3c) ii)

resize → [0] n = 0
resize → [1] n = 1
resize → [1,1] n = 2
resize → [1,1,1] n = 3
resize → [1,1,1,1] n = 4
⋮
resize → [1,1,1,...,1] n = n-1
resize → [1,1,1,...,1,1] n = n

n levels

The average resize costs $O(n/2) = O(n)$
overall runtime is $\Omega(n) \cdot \Omega(n/2)$
$= \Omega(n^2)$

4b)

n levels

[a b c d a b c d ...] found = {a: 1}
[a b c d a b c d ...] found = {a: 1, b: 1}
[a b c d a b c d ...] found = {a: 1, b: 1, c: 1}
⋮
[a b c d a b c d ... d] found = {a: ?, b: ?, ...}

$O(1)$ $O(1)$

total runtime: $O(1) \cdot O(1) = O(1)$

5a)

[a, b, c, d, a, b, c, d] remove a

[b, c, d, a, b, c, d]

n levels

[b, c, d, a, b, c, d]

[b, c, d, b, c, d ...]

~~remove() is O(n)~~

remove() is O(n)

worst case it occurs n times

runtime is $O(n)(O(n)) = O(n^2)$

5c)

n levels, O(1) time per level

[a, a, a, ... b, b, b] remove(a)

[a, a, a, ... b, b, b] count = 1) O(1)

[a, a, a ... b, b, b] count = 2) O(n)

O(n) overall

[a, a, a, .. b, b, b] count = $\frac{n}{2}$

n levels.

copying ints is O(1)

total: O(n)

[b, a, a ... b, b, b]

[b, b, a, .. b, b, b]

[b, b, b ... b, b, b] see 3c

Popping n elements is O(n)

total: $O(n) + O(n) + O(n) = O(n)$