Hashmap<String, Integer> hm = new HashMap<>();

key value

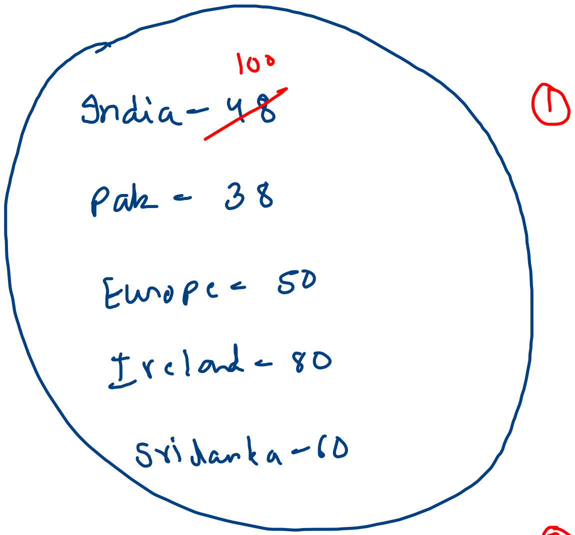
(i) key - value pair

(ii) unordered

(iii) unique - key

③ hm.containsKey

- key ✓ → true
- key ✗ → false



HM

① hm.put("srilanka", 60);

hm.put("India", 100);

put

- key ✓ → updation
- key ✗ → insert

② hm.get("India") -> 100

hm.get("England") -> null

get

- key ✓ → value
- key ✗ → null

④ hm.keySet() -> ["India", "Pak", "Europe", "Srilanka", "Ireland"]

⑤ hm.remove(key)

- key ✓ → remove
- key ✗ → do nothing

highest

freq char

str : f c a b a b c a d e g a g g

f - 1

c - 2

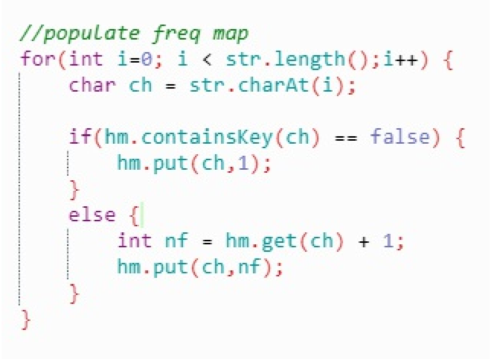
a - 4

b - 2

d - 1

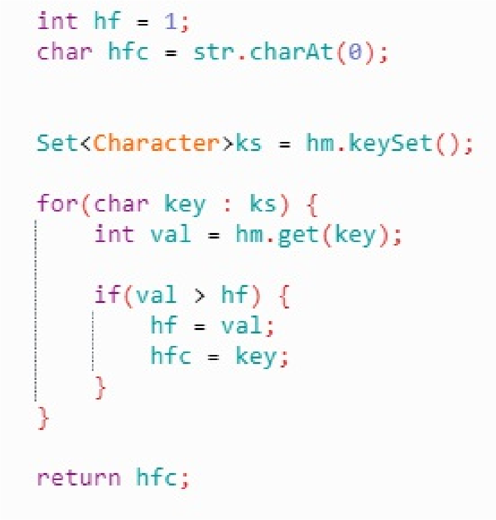
g - 3

hfc -> a



```
//populate freq map
for(int i=0; i < str.length();i++) {
    char ch = str.charAt(i);

    if(hm.containsKey(ch) == false) {
        hm.put(ch,1);
    }
    else {
        int nf = hm.get(ch) + 1;
        hm.put(ch,nf);
    }
}
```



```
int hf = 1;
char hfc = str.charAt(0);


Set<Character>ks = hm.keySet();

for(char key : ks) {
    int val = hm.get(key);

    if(val > hf) {
        hf = val;
        hfc = key;
    }
}

return hfc;
```

j c a b a b c a d e g a g g

j - 1

c - ~~1~~ 2

a - ~~1~~ ~~2~~ ~~3~~ 4

b - ~~1~~ 2

d - 1

e - 1

g - ~~1~~ ~~2~~ 3

hf = ~~1~~ ~~2~~ 4

hfc = ~~j~~ ~~c~~ a

ks : [j, c, a, b, d, g, e]

get common elements

a1 : 1 1 2 3 3 5

a2 : 1 3 2 3 6

a1 → hmap

~~1 - 2~~

~~2 - 1~~

~~3 - 2~~

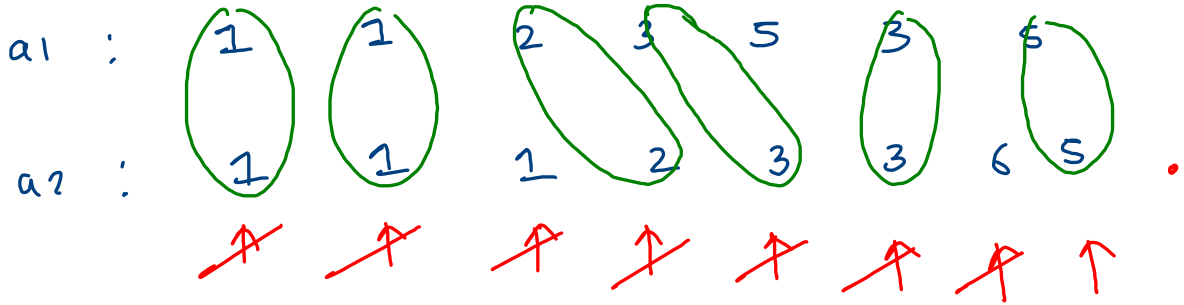
5 - 1

1

3

2

get common elements - 2



a1 : 1 1 2 3 5 3 5

a2 : 1 1 1 2 3 3 6 5

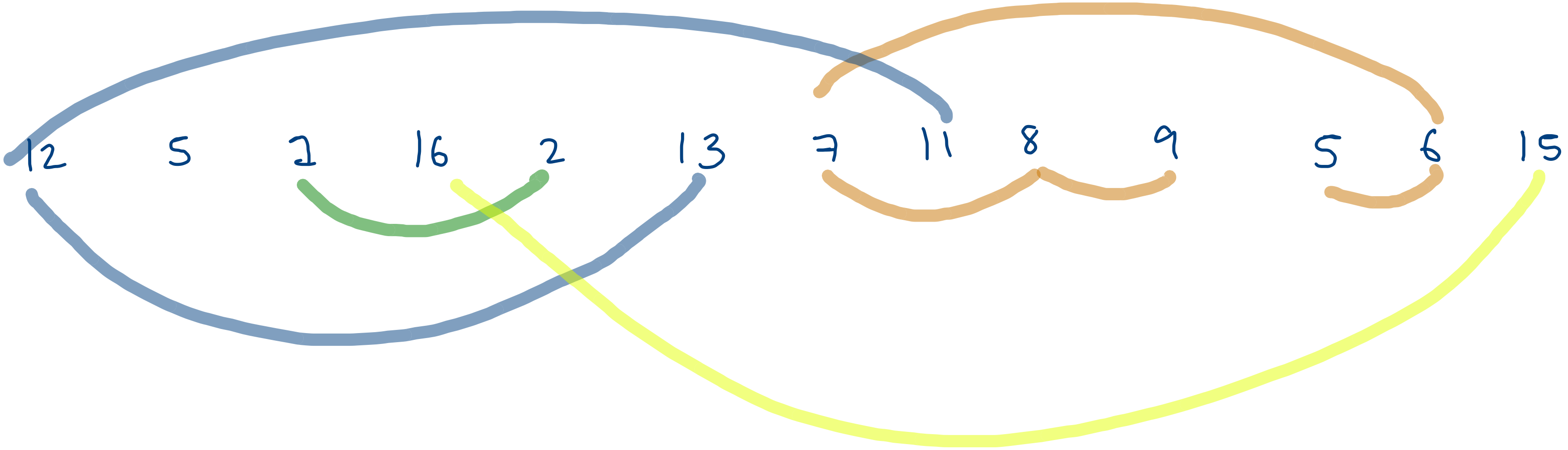
gce : 1 1 2 3 3 5

1 - ~~2~~ ~~1~~ 0

2 - ~~1~~ 0

3 - ~~2~~ ~~1~~ 0

5 - ~~2~~ 1



5

1 2

→ 5 6 7 8 9 (longest consec. sequnce)

11 12 13

15 16

12 5 1 16 2 13 7 11 8 9 5 6 15

~~T~~ ~~T~~ ~~T~~ ~~T~~ ~~T~~ ~~T~~ ~~T~~ ~~T~~ ~~T~~ ~~T~~ ~~T~~ ~~T~~ T

Integer — boolean (seq. start or not)

5 6 7 8 9

12 — ~~T~~ F

5 — T

1 — T

16 — ~~T~~ F

2 — ~~T~~ F

13 — ~~T~~ F

7 — ~~T~~ F

11 — T

8 — ~~T~~ F

9 — ~~T~~ F

6 — ~~T~~ F

15 — T

5 st

len -> ~~1~~ ~~2~~ ~~3~~ 4 5

1 st

len -> ~~1~~ 2

15 st

len -> ~~1~~ 2

11 st

len -> ~~1~~ ~~2~~ 3

12 5 1 16 2 13 7 11 8 9 5 6 15

~~↑~~ ~~↑~~ ~~↑~~ ~~↑~~ ~~↑~~ ~~↑~~ ~~↑~~ ~~↑~~ ~~↑~~ ~~↑~~ ~~↑~~ ~~↑~~ ↑

Integer - boolean (seq. st or not)

12 - ~~T~~ F
✓ 5 - T
✓ 1 - T
16 - ~~T~~ F
2 - ~~T~~ F

13 - ~~T~~ F
7 - ~~T~~ F
✓ 11 - T
8 - ~~T~~ F
9 - ~~T~~ F

6 - ~~T~~ F
15 - T

① - each element is a seq. start.

② - Real seq. start

③ - find longest consec seq.

5-st	1-st	11-st	15-st
5,6, 7, 8, 9	1 2	11 12 13	15-16
(5)	(2)	(3)	(2)

12 5 1 16 2 13 7 11 8 9 5 6 15

~~↑~~ ~~↑~~ ~~↑~~ ~~↑~~ ~~↑~~ ~~↑~~ ~~↑~~ ~~↑~~ ~~↑~~ ~~↑~~ ~~↑~~ ~~↑~~ ↑

12 - ~~T~~ F

✓ 5 - T

✓ 1 - T

16 - ~~T~~ F

2 - ~~T~~ F

13 - ~~T~~ F

7 - ~~T~~ F

✓ 11 - T

8 - ~~T~~ F

9 - ~~T~~ F

6 - ~~T~~ F

✓ 15 - T

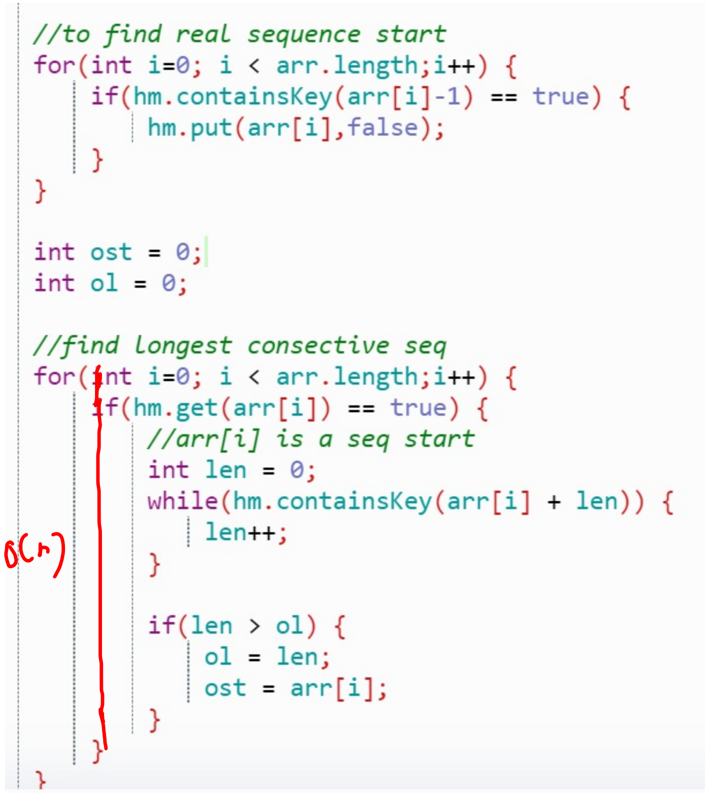
ost = ~~0~~ 5

ol = ~~0~~ 5

len = ~~0~~ ~~1~~ 2

✓ 5 × 6 × 7 × 8 × 9 ✓ 1 × 2

✓ 11 × 12 × 13 ✓ 15 × 16



```
//to find real sequence start
for(int i=0; i < arr.length;i++) {
    if(hm.containsKey(arr[i]-1) == true) {
        hm.put(arr[i],false);
    }
}

int ost = 0;
int ol = 0;

//find longest consective seq
for(int i=0; i < arr.length;i++) {
    if(hm.get(arr[i]) == true) {
        //arr[i] is a seq start
        int len = 0;
        while(hm.containsKey(arr[i] + len)) {
            len++;
        }

        if(len > ol) {
            ol = len;
            ost = arr[i];
        }
    }
}
```

O(n)