In [0]:

```
import numpy as np
import random

l = list(range(100))
random.shuffle(l)

l
```

```
Out[0]: [45,
          19,
          44,
          12,
          63,
          27,
          1,
          35,
          56,
          40,
          59,
          78,
          97,
          28,
          51,
          90,
          77,
          34,
          50,
          29,
          10,
          76,
          9,
          15,
          16,
          69,
          67,
          42,
          64,
          54,
          6,
          83,
          61,
          53,
          49,
          39,
          99,
          55,
          48,
          41,
          75,
          66,
          24,
          5,
          30,
          26,
          92,
          14,
          89,
          73,
          7,
          2,
          65,
          91,
          84,
          98,
          60,
93,
62,
95,
87,
0,
22,
80,
79,
71,
70,
25,
68,
85,
4,
31,
11,
23,
52,
72,
96,
33,
43,
47,
36,
18,
20,
3,
58,
86,
82,
8,
21,
88,
13,
17,
57,
74,
46,
81,
38,
32,
94,
37]
```

In [0]:

```
# search for an elemnt q in the list: O(n) where n is the length of the list
q = 31
isFound=False;
for ele in l:
    if ele==31:
        print("Found")
        isFound=True
        break;
if isFound == False:
    print("Not Found")
```

```
Found
```

In [0]:

```
#What if the list is sorted? Can we search faster?
# Show O(log n)

import math

#Source: http://www.geeksforgeeks.org/binary-search/
#Returns index of x in arr if present, else -1
def binarySearch (arr, l, r, x):

    # Check base case
    if r >= l:

        mid = l + math.floor((r - l)/2)

        # If element is present at the middle itself
        if arr[mid] == x:
            return mid

        # If element is smaller than mid, then it can only
        # be present in left subarray
        elif arr[mid] > x:
            return binarySearch(arr, l, mid-1, x)

        # Else the element can only be present in right subarray
        else:
            return binarySearch(arr, mid+1, r, x)

    else:
        # Element is not present in the array
        return -1


l.sort();
arr = l;
q =31;
binarySearch(arr,0,len(arr)-1,q)
```

Out[0]:

```
31
```

In [0]:

```
# Find elements common in two lists:
l1 = list(range(100))
random.shuffle(l1)


l2 = list(range(50))
random.shuffle(l2)

# find common elements : O(n*m)
cnt=0;
for i in l1:
    for j in l2:
        if i==j:
            print(i)
            cnt += 1;
print("Number of common elements:", cnt)
```

```
43
44
46
38
47
35
45
36
34
41
4
28
18
19
40
29
12
7
33
23
1
30
42
20
8
6
24
22
14
27
48
26
13
39
17
11
0
25
37
10
49
2
9
31
21
32
16
3
5
15
Number of common elements: 50
```

In [0]:

```
# Find elements common in two lists:
l1 = list(range(100))
random.shuffle(l1)


l2 = list(range(50))
random.shuffle(l2)

# find common elemnts in lists in O(n) time and O(m) space if m<n

## add all elements in the smallest list into a hashtable/Dict: O(m) space
smallList = {}
for ele in l2:
    smallList[ele] = 1; # any value is OK. Key is important

# Now find common element
cnt=0;
for i in l1:
    if smallList.get(i) != None: # search happens in constant time.
        print(i);
        cnt += 1;
print("Number of common elements:", cnt)
```

```
21
44
17
30
23
2
15
37
24
29
45
10
32
7
19
36
47
14
41
16
5
42
34
33
39
0
31
1
27
35
22
8
46
20
18
13
25
3
26
12
40
49
4
6
11
9
38
28
48
43
Number of common elements: 50
```