Binary search - on sorted list - LogN

```
def bsearch(arr, target):
        left, right = 0,
len(arr) - 1
        while left <= right:
                mid = (left +
right) // 2
                val = arr[mid]
                if val == target:
                        return
mid
                elif val >
target:
                        right =
mid - 1
                else:
                       left =
mid + 1
        return -1
```

Heap - LogN

```
import heapq
def heap_m in( arr):
        hea pq.h ea pif y(arr)
        return arr[0]
def heap_m ax( arr):
        heaparr = []
       for val in arr:
                hea pq.h ea -
ppu sh( hea parr, -1 * val)
        return -1 * heaparr[0]
```

DFS - preorder - N

```
def dfs(root, target):
       def search (node):
                if not node:
                        return
                if node.val ==
target:
                        return
node
                left = search -
(no de.l eft)
                if left:
                        return
left
                 right = search -
(no de.r ight)
                if right:
                        return
right
        return search (root)
```

Collections

defaultdict	defaultdict(default_factory-=None)
namedtuple	namedtuple('Point', ['x', 'y'])
deque	deque([iterable[, maxlen]])
	append(x)
	appendleft(x)
	pop()
	popleft()
Counter	Counter([iterable-or-map-ping])
	most_common(n)
	subtract([iterable-or-ma-pping])

BFS - N

```
def bfs(root, target):
        queue = [root]
        while queue:
                node = queue.p -
op()
                if not node:
                        con -
tinue
                if node.val ==
target:
                        return
node
                 que ue.a pp -
end (no de.l eft)
                 que ue.a pp -
end (no de.r ight)
        return None
```

Python Slices

Slice	a[start:s-top:step]
Elements after X	a[x:]
First X elements	a[:x]
Last X elements	a[-x]
Elements from X to Y not inclusive of Y	a[x:y]
Reverse List	a[::-1]

By **dkarp**
cheatography.com/dkarp/

Published 15th September, 2021.
Last updated 15th September, 2021.

Sponsored by **CrosswordCheats.com**
Learn to solve cryptic crosswords!
http://crosswordcheats.com