

## Heap

- It will work as a priority queue . It will gives you the highest priority element

h = [ 10 , 20 , 50 , 40 , 60 , 30 , 70 ]

- Smaller the number will be the highest priority so 10 has the highest priority and 70 will be the least priority . That's means the larger the number will be the least priority
- **heappush** : -

```
>>> import heapq
>>> H = []
>>>
>>> heapq.heappush(H,20)
>>> H
[20]
>>> heapq.heappush(H,50)
>>> H
[20, 50]
>>> heapq.heappush(H,10)
>>> H
[10, 50, 20]
>>> heapq.heappush(H,40)
>>> heapq.heappush(H,30)
>>> heapq.heappush(H,60)
>>> H
[10, 30, 20, 50, 40, 60]
>>>
```

For this you have to import the module heapq . By using heappush we can add the elements .

- **heappop** :

```
>>> heapq.heappop(H)
10
>>> H
[20, 30, 60, 50, 40]
>>> heapq.heappop(H)
20
>>> H
[30, 40, 60, 50]
```

- heappop it will remove the highest priority ( H )
- **heapify** :
- If we have taken a list example  
L = [50,30,60,40,70,20,10]

As you can see that this list is not in the form of heap.

- So heapify will convert the list into heap form

**heapq.nlargest and heapq.nsmallest :**

- This method is use to find the largest and smallest among the heap

`heapq.nlargest( 2 , L )`

This means two largest number

`heapq.nsmallest( 3,L)`

This means give 3 smallest number present in the List