

Next fit , First fit Decreasing

```
1 def nextfit(weight, c):
2     bins = 0
3     bin_list = [[]]
4     remainder = c
5     for i in range(len(weight)):
6         # check if the current items fits in the previous bin
7         if remainder >= weight[i]:
8             remainder = remainder - weight[i]
9             bin_list[bins].append(weight[i])
10        else:
11            bin_list.append([])
12            bins += 1
13            remainder = c - weight[i]
14            bin_list[bins].append(weight[i])
15    return bin_list
16
17
18 ▼ def firstfit_decreasing(weight, c):
19     # sort the items in descending order
20     weight = sorted(weight, reverse=True)
21     bins = 0
22     bin_list = [[]]
23     remainder = c
24 ▼    for i in range(len(weight)):
25 ▼        if remainder >= weight[i]:
26            remainder = remainder - weight[i]
27            bin_list[bins].append(weight[i])
28 ▼        else:
29            bin_list.append([])
30            bins += 1
31            remainder = c - weight[i]
32            bin_list[bins].append(weight[i])
33    return bin_list
34
35
```

```

36 def display(list1): # display the items bin wise
37     length = len(list1)
38     print(f"{length}\nitems will be fit in below manner : ")
39     for i in range(len(list1)):
40         print(f"Bin-{i+1} --> ", end='')
41         print(*list1[i])
42
43
44 if __name__ == '__main__':
45     weight = [2, 5, 4, 7, 1, 3, 8]
46     c = 10
47     result_nextfit = nextfit(weight, c)
48     print("Number of bins required in Next Fit :", end=' ')
49     display(result_nextfit)
50     print("-----")
51     result_firstfit_Dec = firstfit_decreasing(weight, c)
52     print("Number of bins required in First Fit Decreasing :", end=' ')
53     display(result_firstfit_Dec)

```

Output

Number of bins required in Next Fit : 5
items will be fit in below manner :

```

Bin-1 --> 2 5
Bin-2 --> 4
Bin-3 --> 7 1
Bin-4 --> 3
Bin-5 --> 8

```

Number of bins required in First Fit Decreasing : 4
items will be fit in below manner :

```

Bin-1 --> 8
Bin-2 --> 7
Bin-3 --> 5 4
Bin-4 --> 3 2 1

```

Repl Closed

Randomized Quick Sort

```
1  import random
2
3
4  def quicksort(arr, start, stop):
5      if(start < stop):
6          # pivotindex is the index where the pivot lies in the array
7          pivotindex = select_pivot(arr, start, stop)
8          quicksort(arr, start, pivotindex)
9          quicksort(arr, pivotindex + 1, stop)
10
11 # This function generates random pivot, swaps the first element with the pivot
12
13
14 def select_pivot(arr, start, stop):
15     randpivot = random.randrange(start, stop)
16     print("-----")
17     print(f"Let the pivot be - {array[randpivot]}")
18     arr[start], arr[randpivot] = arr[randpivot], arr[start]
19     return partition(arr, start, stop)
20
21
22 def partition(arr, start, stop):
23     pivot = start # pivot
24     i = start - 1
25     j = stop + 1
26     while True:
27         while True:
28             i = i + 1
29             if arr[i] >= arr[pivot]:
30                 break
31         while True:
32             j = j - 1
33             if arr[j] <= arr[pivot]:
34                 break
35         if i >= j:
36             return j
37         arr[i], arr[j] = arr[j], arr[i]
38         print(f"Swapping the elements {arr[i]} & {arr[j]} ==> {array}")
39
40
41 if __name__ == '__main__':
42     array = [10, 7, 8, 9, 1, 5, 18, 6, 36, 4, 15]
43     print(f"Unsorted array : {array}")
44     temp_array = array
45     quicksort(array, 0, len(array) - 1)
46     print("-----")
47     print(f"Sorted array :{array}")
48
```

Output

```
Unsorted array : [10, 7, 8, 9, 1, 5, 18, 6, 36, 4, 15]
```

```
-----  
Let the pivot be - 1  
-----
```

```
Let the pivot be - 5
```

```
Swapping the elements 4 & 5 ==> [1, 4, 8, 9, 10, 7, 18, 6, 36, 5, 15]  
-----
```

```
Let the pivot be - 7
```

```
Swapping the elements 5 & 7 ==> [1, 4, 5, 9, 10, 8, 18, 6, 36, 7, 15]  
-----
```

```
Let the pivot be - 10
```

```
Swapping the elements 7 & 10 ==> [1, 4, 5, 7, 9, 8, 18, 6, 36, 10, 15]
```

```
Swapping the elements 6 & 9 ==> [1, 4, 5, 7, 6, 8, 18, 9, 36, 10, 15]  
-----
```

```
Let the pivot be - 7
```

```
Swapping the elements 6 & 7 ==> [1, 4, 5, 6, 7, 8, 18, 9, 36, 10, 15]  
-----
```

```
Let the pivot be - 18
```

```
Swapping the elements 15 & 18 ==> [1, 4, 5, 6, 7, 15, 8, 9, 36, 10, 18]
```

```
Swapping the elements 10 & 36 ==> [1, 4, 5, 6, 7, 15, 8, 9, 10, 36, 18]  
-----
```

```
Let the pivot be - 15
```

```
Swapping the elements 10 & 15 ==> [1, 4, 5, 6, 7, 10, 8, 9, 15, 36, 18]  
-----
```

```
Let the pivot be - 10
```

```
Swapping the elements 9 & 10 ==> [1, 4, 5, 6, 7, 9, 8, 10, 15, 36, 18]  
-----
```

```
Let the pivot be - 9
```

```
Swapping the elements 8 & 9 ==> [1, 4, 5, 6, 7, 8, 9, 10, 15, 36, 18]  
-----
```

```
Let the pivot be - 36
```

```
Swapping the elements 18 & 36 ==> [1, 4, 5, 6, 7, 8, 9, 10, 15, 18, 36]  
-----
```

```
Sorted array : [1, 4, 5, 6, 7, 8, 9, 10, 15, 18, 36]
```

```
***Repl Closed***
```