

**AA - Experiment 3 - Randomised Quicksort**

60004200155

Jigar Siddhpura

C22

Experiment 3 - Randomised quick sortAim: To implement randomised quick sortTheory:

1. Randomization is a powerful technique used in advance algos to overcome worst case scenarios, that can plague deterministic algos.
2. It introduces a degree of uncertainty by incorporating random choices during execution.
3. The random choice often have a significant impact on average case performance making algorithm more efficient across a wider range of inputs.
4. Quicksort is a efficient divide & conquer sorting algorithm known for its efficient average case performance  $O(\log n \cdot n)$ , however, the worst case is  $O(n^2)$ .
5. Randomized quick sort solves this problem by introducing randomization in pivot selection.

Working:

1. Random pivot selection - Instead of fixed pivot, random pivot is chosen, various method exist for random selection.
2. Partitioning - algo partition the data around chosen pivot placing element less than pivot to its left & elements greater than or equal to pivot its right.
3. Recursion - Recursively sort left & right.

- Code contains 2 algos - Normal QS & Randomised QS.  
In deterministic it takes a fixed indexed element as pivot.
- key difference :- Normal one leads to worst case for ascending ordered array but in other it breaks potential in input & on average subarrays created during partition are more balanced.
- C1 & C2 took the no. of variables comparisons in both algos respectively which is analyzed later.

Conclusion — Hence, randomized QS was implemented with normal one. For normal, no. of comparisons are 79800 & for randomized QS it's 38203 for already sorted arr of length 400.

## **CODE :**

```
import random
```

```
c1, c2 = 0, 0
```

```
def randomized_qs(arr):  
    global c1  
    if len(arr) <= 1:  
        return arr  
    else:  
        pivot = random.choice(arr)  
        left = []  
        right = []  
        for i in range(len(arr)):  
            if arr[i] < pivot:  
                left.append(arr[i])  
                c1 += 1  
            elif arr[i] > pivot:  
                right.append(arr[i])  
                c1 += 1  
        return randomized_qs(left) + [pivot] + randomized_qs(right)
```

```
def quicksort(arr):  
    global c2  
    if len(arr) <= 1:  
        return arr  
    else:  
        pivot = arr[0]  
        left = []  
        right = []  
        for i in range(1, len(arr)):  
            if arr[i] < pivot:  
                left.append(arr[i])  
                c2 += 1  
            else:  
                right.append(arr[i])  
                c2 += 1  
        return quicksort(left) + [pivot] + quicksort(right)
```



```
arr = [i for i in range(400)]
```

```
print('Normal Quicksort')
```

```
print("Sorted Array:", quicksort(arr))
```

```
print("Number of Comparisons:", c2)
```

```
print("Randomized QS")
```

```
print("Sorted Array:", randomized_qs(arr))
```

```
print("Number of Comparisons:", c1)
```

## OUTPUT :

```
PS D:\SEM-6\AA\EXPERIMENTS> python -u "d:\SEM-6\AA\EXPERIMENTS\randomized_qs.py"
Normal Quicksort
Sorted Array: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37,
, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 7
6, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111,
112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142,
143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173,
174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204,
205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235,
236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266,
267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297,
298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328,
329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359,
360, 361, 362, 363, 364, 365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376, 377, 378, 379, 380, 381, 382, 383, 384, 385, 386, 387, 388, 389, 390,
391, 392, 393, 394, 395, 396, 397, 398, 399]
Number of Comparisons: 79800
Randomized QS
Sorted Array: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37,
, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 7
6, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111,
112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142,
143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173,
174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204,
205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235,
236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266,
267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297,
298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328,
329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359,
360, 361, 362, 363, 364, 365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376, 377, 378, 379, 380, 381, 382, 383, 384, 385, 386, 387, 388, 389, 390,
391, 392, 393, 394, 395, 396, 397, 398, 399]
Number of Comparisons: 3823
PS D:\SEM-6\AA\EXPERIMENTS>
```