Quick Sort

Prepared By

Dr. V. Elizabeth Jesi

Department of IT

Introduction

- An efficient sorting algorithm
- Based on partitioning of array of **data** into smaller arrays
- Developed by British computer scientist Tony Hoare in 1959 and published in 1961
- Two or three times faster than other sorting algorithms
- Uses divide and conquer strategy
- Quicksort is a comparison sort
- Sometimes called **partition-exchange sort**

Three operations performed in quick sort

1. **Divide:** Select a 'pivot' element from the array and partition the other elements into two sub-arrays, according to whether they are less than or greater than the pivot.
2. **Conquer :** Recursively sort the two sub arrays
3. **Combine** : Combine all the sorted elements in a group to form a list of sorted elements.

How Quick sort works

- Select a PIVOT element from the array
- You can choose any element from the array as the pivot element.
- Here, we have taken the first element of the array as the pivot element.

34	42	16	71	9	22	53	13

- The elements smaller than the pivot element are put on the left and the elements greater than the pivot element are put on the right.

9	18	16	22	**34**	71	42	53

- Now sort the left and right arrays recursively.

Algorithm Quicksort

```
Quicksort(A, L,H)
{
// Array to be soerrted
// L – lower bound of array
// H- upper bound of array

If (L<H)
{
        P=Partition(A,L,H)
        Quicksort(A,L,P)
        Quicksort(A,P+1,H)
}
}
```

Algorithm Partition

```
Partition(A,L,H)
{
        pivot = A[L]
        i=L, j=H+1
    repeat
         repeat   i=i+1  until A[i]>=pivot
         repeat   j=j-1    until A[j]<=pivot
         swap(a[i],a[j])
    until (i>j)
    swap(A[L],A[j])
}
```

Method with Example (Partition)

- Numbers to be sorted

34	53	16	71	9	22	42	18

- Select 34(first element) as the pivot element

34	53	16	71	9	22	42	18

Find the element > 34 from the left, let its index be i

Find the element<34 from the right, let its index be j

34	**53**	16	71	9	22	42	**18**

34	**18**	16	71	9	22	42	**53**

- Do this process until the index I is greater than index j.

34	18	16	**71** **i=4**	9	**22** **j=6**	42	53

34	18	16	**22**	9	**71**	42	52

34	18	16	22	**9** **j=5**	**71** **i=6**	42	53

- I becomes > j , so exchange the pivot element and A(j)

9	18	16	22	**34**	71	42	53

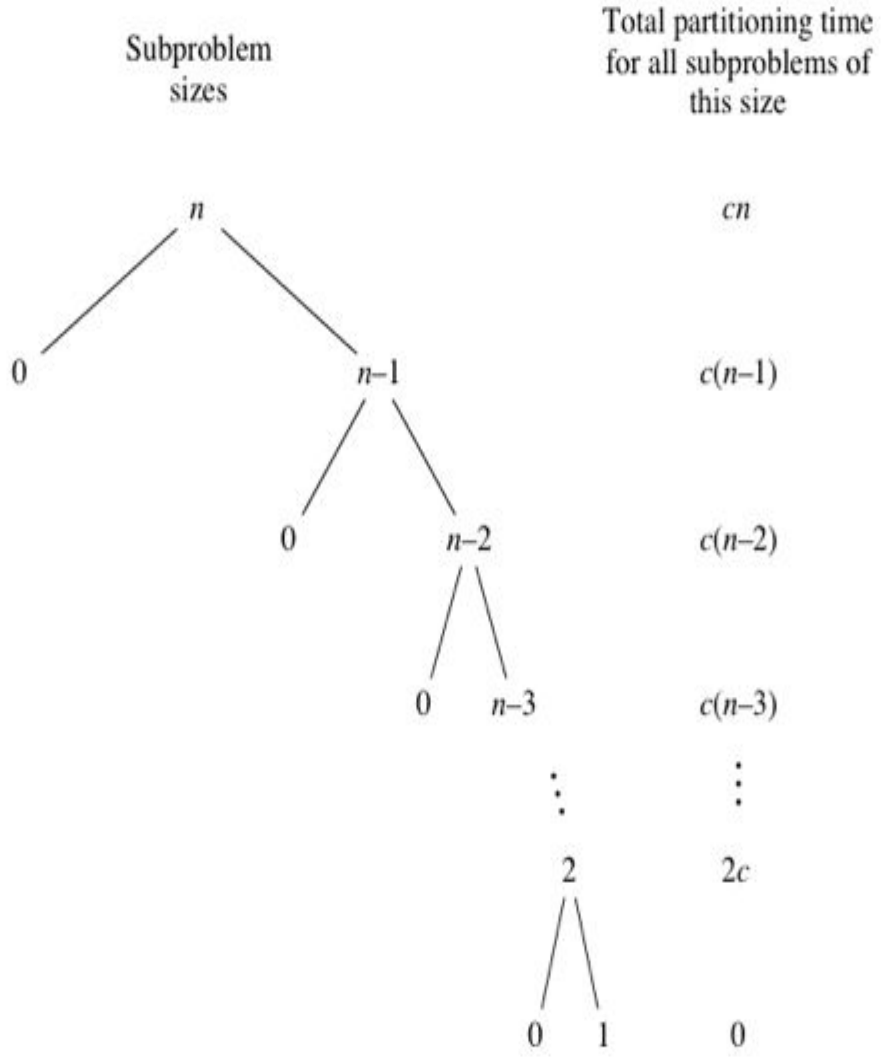
- Note all the elements to the left of pivot is lesser and the elements to the right of pivot is larger than the pivot

Analysis of Quick Sort Algorithm

- If pivot element is either the largest or the smallest element
- Then one partition will have 0 elements and the other partition will have (n-1) elements.
- Eg : if pivot element is the largest element in the array. We will have all the elements except the pivot element in the left partition and no elements in the right partition
- The recursive calls will be on arrays of size 0 and (n-1)

Quick Sort - Worst-case running time

- Let the original call takes cn time for some constant c.
- Recursive call on n-1 elements takes c(n-1)times
- Recursive call on n-2 elements takes time c(n-2) times, and so on.
- Recursive call on 2 elements will take 2c times
- Finally recursive call on 1 element will take no time.



- Sum up the partitioning times, we get

  $cn + c(n-1) + c(n-2) + \ldots + 2c$

  $= c(n + (n-1) + (n-2) + \ldots + 2)$

  $= c((n+1)(n/2) - 1)$

[Note : $1+2+\ldots+(n-1)+n=(n+1)n/2$

So $2+\ldots+(n-1)+n=((n+1)n/2)-1$]

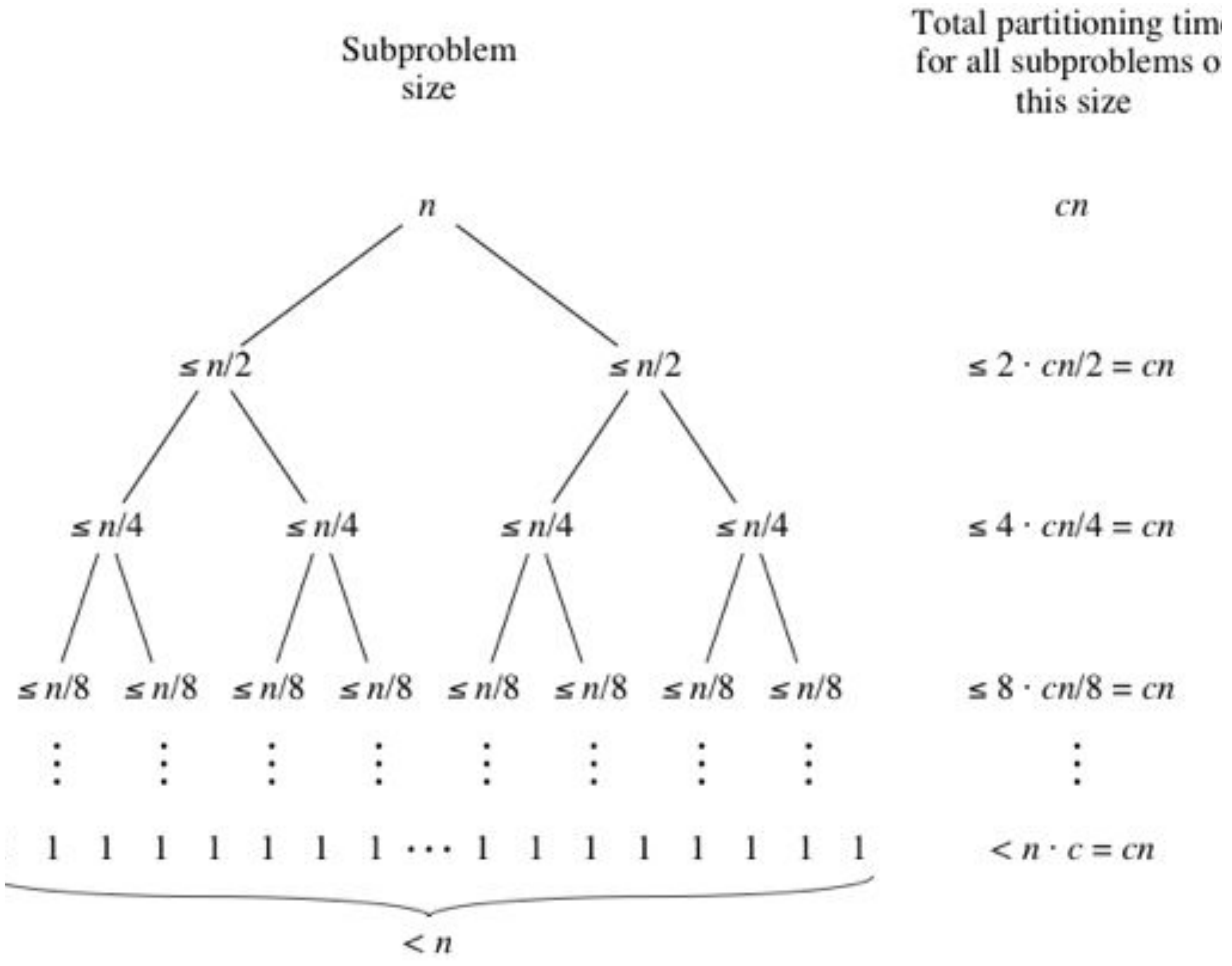
quicksort's worst-case running time is $\Theta(n^2)$

Quick Sort : Best-case running time

Occurs when the partitions are evenly partitioned.

quicksort's Best-case running time is

$$\Theta(n \log_2 n)$$



Quick Sort : Average-case running time

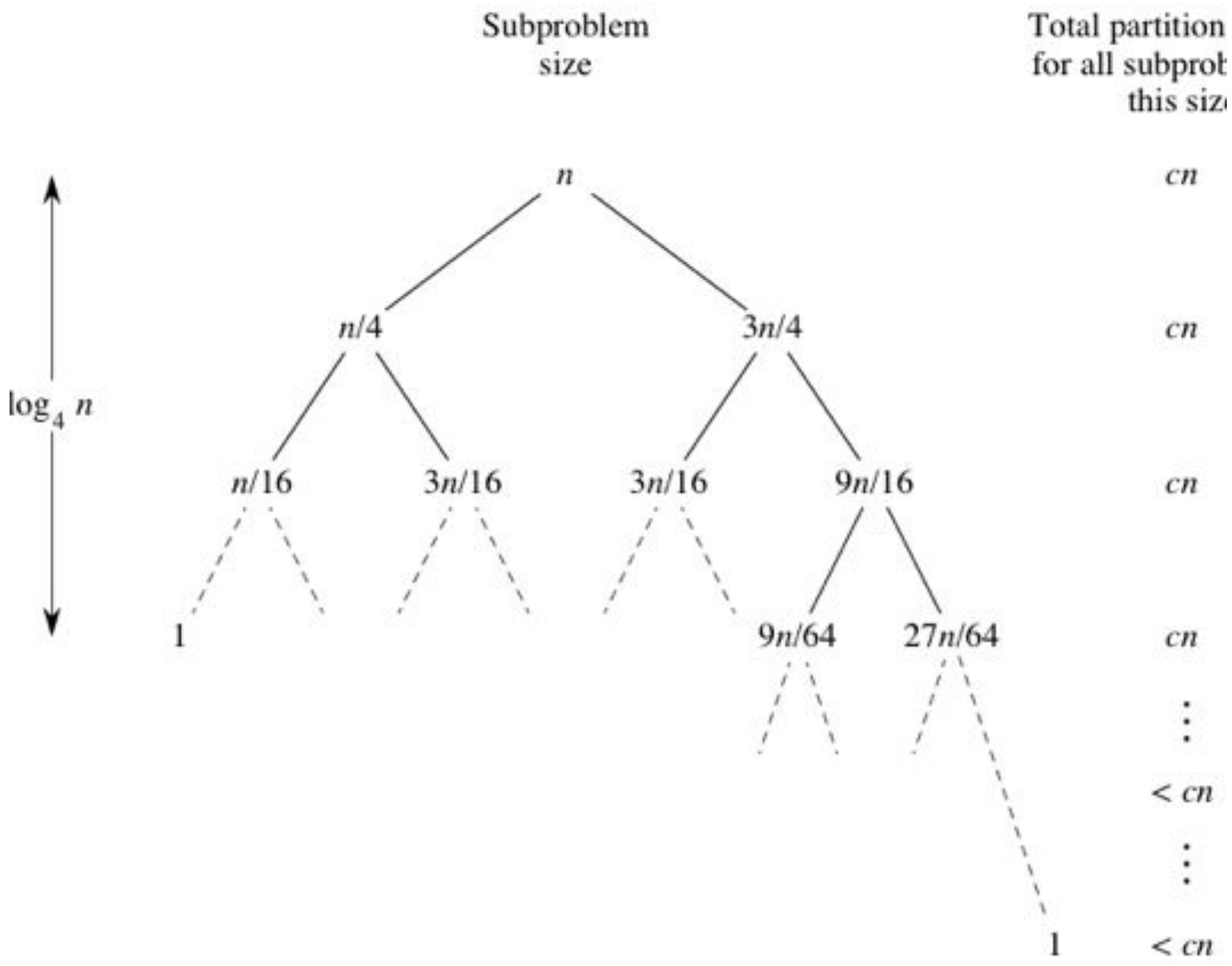
imagine that we don't always get evenly balanced partitions

but that we always get at worst a 3-to-1 split

That is, imagine that each time we partition, one side gets 3n/4 elements and the other side gets n/4 elements

quicksort's Average-case running time is

$$\Theta(n \log_2 n)$$



Home assignment

- Sort the following numbers using Quick sort

- 21, 45, 32, 76, 12, 83, 47, 153, 52
- 75, 34, 64, 82,35,79,12,53,40,61