Randomised Quicksort

Chap 7 from CLRS

Recap

- QuickSort?
- Technique used?
- Workhorse in QuickSort? MergeSort?
- Solving recurrences using recursion tree and Master theorem?

What is Randomised Quicksort?

- Instead of A[r] as the pivot always, choose an element randomly from A[p..r]
- Then exchange this element with A[r] and proceed as before.
- **Advantage?**
- We are now ensuring that the pivot can be any of the r-p+1 elements in the array.

Randomised Partition method

```
RANDOMIZED-PARTITION(A, p, r)
i = RANDOM(p, r)
exchange A[r] with A[i]
return PARTITION(A, p, r)
```

```
PARTITION(A, p, r)
x = A[r]
i = p - 1
for j = p to r - 1
    if A[j] ≤ x
        i = i + 1
        exchange A[i] with A[j]
exchange A[i + 1] with A[r]
return i + 1
```

Randomised Quicksort

RANDOMIZED-QUICKSORT(A, p, r)

if $p < r$
    q = RANDOMIZED-PARTITION(A, p, r)
    RANDOMIZED-QUICKSORT($A, p, q - 1$)
    RANDOMIZED-QUICKSORT($A, q + 1, r$)

Analysis

Already seen:

$$\mathbf{T(n) = T(k) + T(n\text{-}k\text{-}1) + \Theta(n)}$$

Questions:

1. what is the min value of k? What is the solution?
2. what is the maximum value of k? What is the solution?
3. where is Θ(n) coming from?
4. Is the relation valid for randomised-quicksort?

Take-away

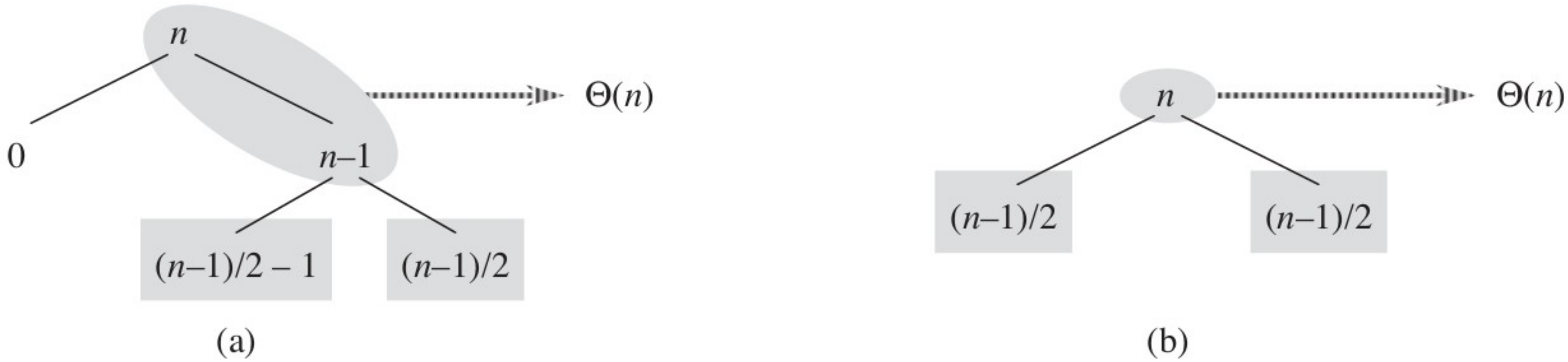
If the split induced by Partition or randomised-partition puts any constant fraction of the elements on one side of the partition, then the recursion tree has height of $\Theta(\log n)$.

Question: For height, Is the base of log equal to 2? If not, what is it?

To think about:

It is highly unlikely that the partitioning will be always ‘best’ or ‘worst’ at every level.

Some splits → well balanced; some splits → highly unbalanced.



To think about (2):

Case ‘a’: subarrays of size 0, n-1/2, n-1/2 -1

Partitioning cost? $\Theta(n) + \Theta(n-1) = \Theta(n)$

Case ‘b’: subarrays of size n-1/2 and n-1/2

Partitioning cost? $\Theta(n)$

Alternates between good and bad levels → expected, therefore average turns out to be O(nlogn)

Some questions on QuickSort

- What is the maximum no. of times two elements are compared to one another in quick sort?
- What is the minimum?
- Is QuickSort in-place?
- Suppose we are sorting an array of eight integers using quicksort, and we have just finished the first partitioning with the array looking like this:

  2 5 1 7 9 12 11 10

- What is the pivot?

Some questions on quicksort (2)

- What will be the recurrence relation for quick sort if the n/4th smallest element is chosen as the pivot?
- Consider the Quicksort algorithm. Suppose there is a procedure for finding a pivot element which splits the list into two sub-lists each of which contains at least one-fifth of the elements. Let T(n) be the number of comparisons required to sort n elements. Then

(a) $T(n) <= 2T(n/5) + n$

(b) $T(n) <= T(n/5) + T(4n/5) + n$

(c) $T(n) <= 2T(4n/5) + n$

(d) $T(n) <= 2T(n/2) + n$

Lab for this week

Programming task

(1)

merge sort

Implement Quicksort for inputs n = 20 to 100.

Sunday (30/01/22) 11:59PM.

Analyze the

The program should report the no. of comparisons.

→ What comparisons?

In best case and worst case. + average case (random inputs)

correctness already done.

(2) Quiz @ 1:30 PM, Friday, 28/01/2022.