Design and Analysis of Algorithms I

QuickSort

Choosing a Good Pivot

QuickSort: High-Level Description

[Hoare circa 1961]

QuickSort (array A, length n)

-If n=1   return

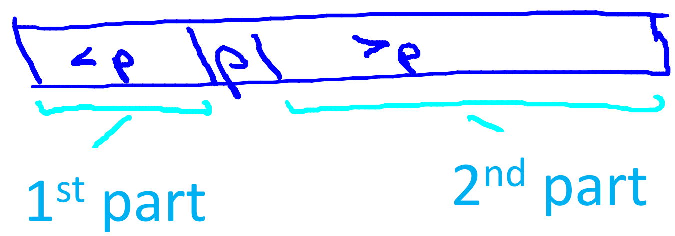
-p = ChoosePivot(A,n)

-Partition A around p

-Recursively sort 1st part

-Recursively sort 2nd part

[currently unimplemented]



The Importance of the Pivot

Q : running time of QuickSort ?

A : depends on the <u>quality of the pivot.</u>

Tim Roughgarden

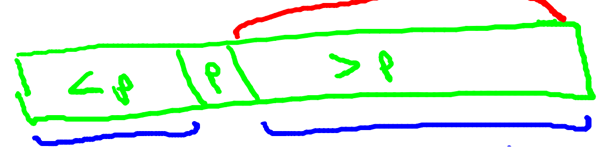
Ter
ver

Suppose we implement QuickSort so that ChoosePivot always selects the first element of the array. What is the running time of this algorithm on an input array that is already sorted?

Recurse on these

- ○ Not enough information to answer question
- ○ $\theta(n)$
- ○ $\theta(n \log n)$
- ○ $\theta(n^2)$

Reason :



empty

Length n-1 (still sorted)

1st n/2 terms are all at least n/2

Runtime :

$$\geq n + (n-1) + (n-2) + \ldots + 1$$
$$= \theta(n^2)$$

Ter
ver

Suppose we run QuickSort on some input, and, magically, every recursive call chooses the median element of its subarray as its pivot. What's the running time in this case?

- Not enough information to answer question
- $\theta(n)$
- $\theta(n \log n)$
- $\theta(n^2)$

<u>Reason</u> : Let T(n) = running time on arrays of size n.

Because pivot = median

choosePivot

partition

Then : $T(n) \leq 2T(n/2) + \theta(n)$

$=> T(n) = \theta(n \log n) \quad [like \quad MergeSort]$

Random Pivots

Key Question : how to choose pivots ? BIG IDEA : RANDOM PIVOTS!

That is : in every recursive call, choose the pivot randomly.
(each element equally likely)

Hope : a random pivot is "pretty good" "often enough".

Intuition : 1.) if always get a 25-75 split, good enough for O(nlog(n))

running time. [this is a non-trivial exercise : prove via recursion tree]
2.) half of elements give a 25-75 split or better

Q : does this really work ?

Average Running Time of QuickSort

<u>QuickSort Theorem</u> : for every input array of length n, the average running time of QuickSort (with random pivots) is O(nlog(n)).

<u>Note</u> : holds for every input. [no assumptions on the data]

- recall our guiding principles !
- "average" is over random choices made by the algorithm (i.e., the pivot choices)