NPTEL MOOC,JAN-FEB 2015
Week 2, Module 8

DESIGN AND ANALYSIS OF ALGORITHMS

Quicksort: Analysis

MADHAVAN MUKUND, CHENNAI MATHEMATICAL INSTITUTE
http://www.cmi.ac.in/~madhavan

Quicksort

- Choose a pivot element
  - Typically the first value in the array
- Partition A into lower and upper parts with respect to pivot
- Move pivot between lower and upper partition
- Recursively sort the two partitions

Analysis of Quicksort

- Partitioning with respect to pivot takes $O(n)$
- If pivot is median
  - Each partition is of size $n/2$
  - $t(n) = 2t(n/2) + n = O(n \log n)$
- Worst case?

Analysis of Quicksort

Worst case

- Pivot is maximum or minimum
  - One partition is empty
  - Other is size n-1
  - $t(n) = t(n-1) + n = t(n-2) + (n-1) + n$
    $= \dots = 1 + 2 + \dots + n = O(n^2)$
- Already sorted array is worst case input!

Analysis of Quicksort

But …

- Average case is O(n log n)
  - Sorting is a rare example where average case can be computed
- What does average case mean?

Quicksort: Average case

- Assume input is a permutation of {1,2,…,n}
  - Actual values not important
  - Only relative order matters
  - Each input is equally likely (uniform probability)
- Calculate running time across all inputs
- Expected running time can be shown O(n log n)

Quicksort: randomization

- Worst case arises because of fixed choice of pivot
  - We chose the first element
  - For any fixed strategy (last element, midpoint), can work backwards to construct $O(n^2)$ worst case
- Instead, choose pivot randomly
  - Pick any index in [0..n-1] with uniform probability
- Expected running time is again $O(n \log n)$

Iterative Quicksort

- Recursive calls work on disjoint segments of array
  - No recombination of results required
- Can use an explicit stack to simulate recursion
  - Stack only needs to store left and right endpoints of interval to be sorted

Quicksort in practice

- In practice, Quicksort is very fast
  - Typically the default algorithm for in-built sort functions
    - Spreadsheets
    - Built in sort function in programming languages