Design and Analysis of Algorithms I

QuickSort

Proof of Correctness

Induction Review

Let P(n) = assertion parameterized by positive integers n.

For us: P(n) is "QuickSort correctly sorts every input array of length n".

How to prove P(n) for all n ≥ 1 by induction:

(1) [base case] directly prove that P(1) holds.

(2) [inductive step] for every n ≥ 2, prove that:

if $\boxed{P(k) \text{ holds for all } k < n}$, then P(n) holds as well.

INDUCTIVE HYPOTHESIS

Tim Roughgarden

Correctness of QuickSort

$P(n)$ = "QuickSort correctly sorts every input array of length n".

Claim: $P(n)$ holds for every $n \geq 1$. [no matter how pivot is chosen]

Proof by induction:

① [base case] every input array of length 1 is already sorted. QuickSort returns the input array, which is correct. ($P(1)$ holds)

② [inductive step] Fix $n \geq 2$. Fix some input array A of length n.

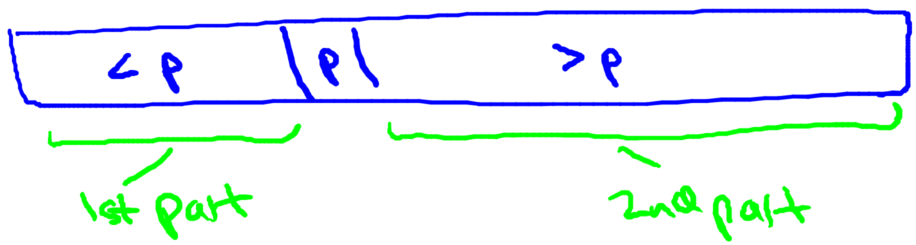
INDUCTIVE HYPOTHESIS

Need to show: if $P(k)$ holds $\forall k < n$, then $P(n)$ holds as well.

Tim Roughgarden

Correctness of QuickSort (con'd)

Recall: QuickSort first partitions A around some pivot p.



Note: pivot winds up in correct position.

Let k_1, k_2 = lengths of 1st, 2nd parts of partitioned array.

Note: $k_1, k_2 < n$

By inductive hypothesis: 1st, 2nd parts get sorted correctly by recursive calls. using $P(k_1)$, $P(k_2)$

So: after recursive calls, entire array correctly sorted. QED!

Tim Roughgarden