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 "Quick Sort correctly sorts every input array of length n"

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

INDUCTIVE HYPOTHESIS

1. [base case] directly prove that P(1) holds.
2. [inductive step] for every n>=2, prove that:
   If P(k) holds for all k<n, then P(n) holds as well.

Correctness of QuickSort

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

<u>Claim :</u> P(n) holds for every n >= 1 [no matter how pivot is chosen]

<u>Proof by induction :</u>

1. [base case] every input array of length 1 is already sorted.
   Quick Sort returns the input array which is correct (so P(1) holds)

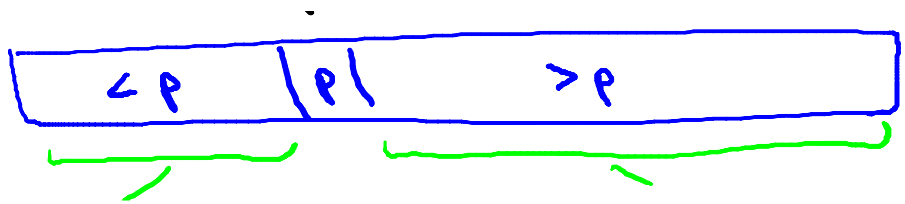
2. [inductive step] Fix n>=2. Fix some input array of length n.

   <u>Need to show</u> : if $\underbrace{\text{P(k) holds for all k < n, then P(n) holds as well.}}_{\text{INDUCTIVE STEP}}$

Tim Roughgarden

Correctness of QuickSort (con'd)

Recall : QuickSort first partitions A around some pivot p.



1^{st} part 2^{nd} part

Note : $k_1, k_2 < n$

Note : pivot winds up in the correct position.
Let k_1, k_2 = lengths of 1^{st}, 2^{nd} parts of partitioned array.

Using $P(k_1)$, $P(k_2)$

By inductive hypothesis : 1^{st}, 2^{nd} parts get sorted correctly by recursive calls. So after recursive calls, entire array correctly sorted.