Quickselect

(1) Define problem

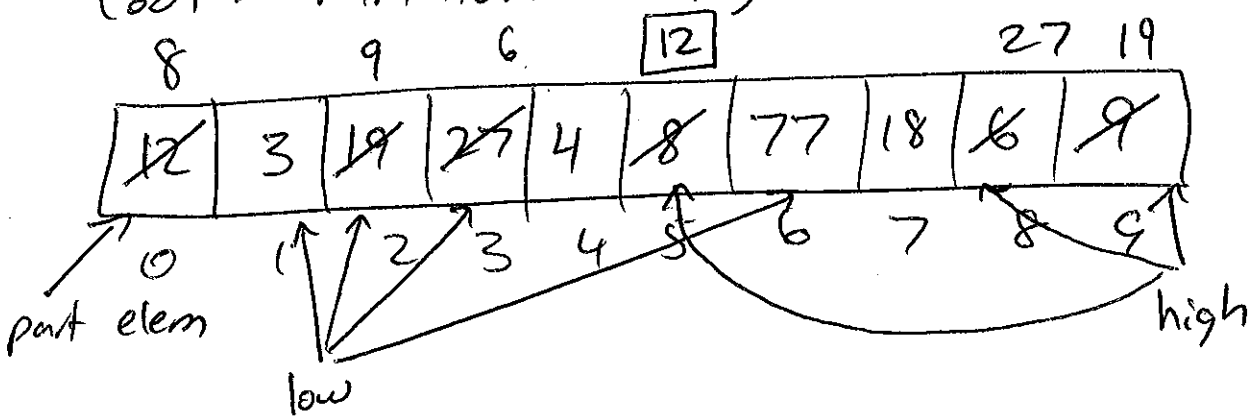
(2) Algorithm to solve it

(3) Avg case run-time

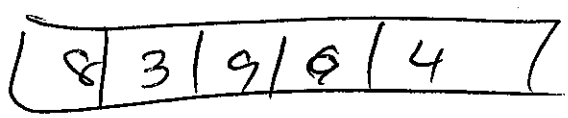
(4) Code up + compare vs theoretical result

Given an array of n items, find the k^{th} smallest.

Tool: Partition Algorithm

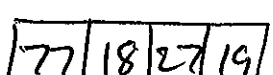


Wanted the 3rd smallest item in the array: Now I want the 3rd smallest in



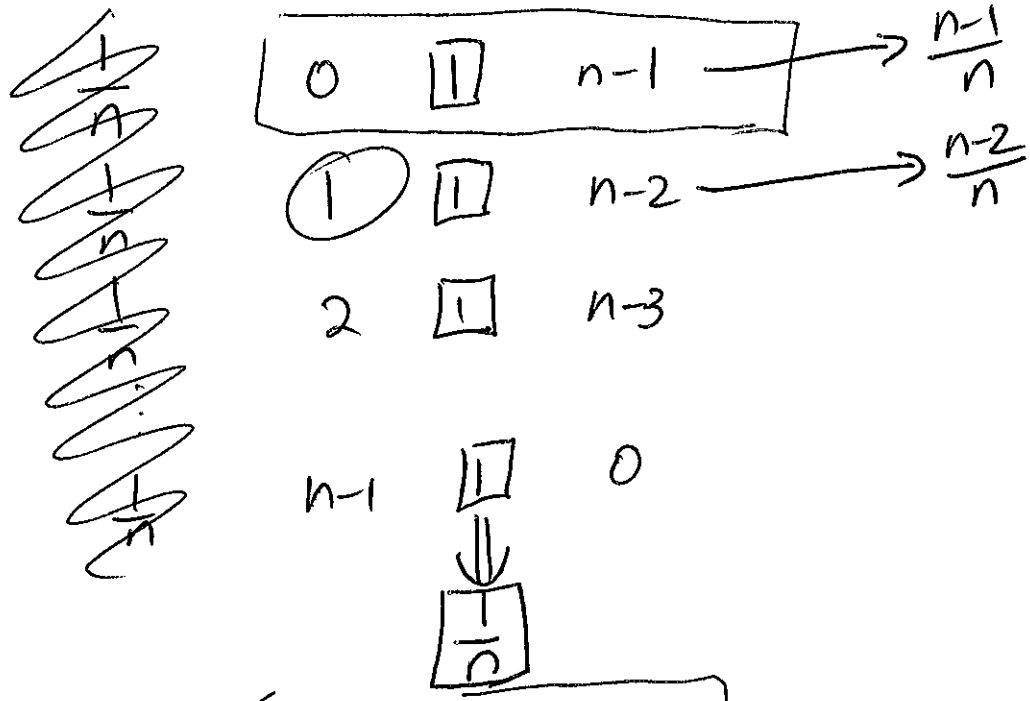
Wanted the 6^{th} smallest item

$\Rightarrow$ I would be done, it's 12.

Wanted the 9^{th} smallest:  (3rd smallest)

Avg Case analysis

Let $T(n)$ = avg case run-time Quick Select of n items



$$nT(n) = \frac{1}{n}(0) + \frac{n-1}{n}T(n-1) + n$$

$$= \frac{1}{n}T(1) + \frac{n-2}{n}T(n-2) + n$$

$$= \frac{2}{n}T(2) + \frac{n-3}{n}T(n-3) + n$$

$$= \frac{3}{n}T(3) + \frac{n-4}{n}T(n-4) + n$$

$$\vdots$$

$$= \frac{n-1}{n}T(n-1) \qquad + n$$

Idea #1

$$nT(n) = 2\left[\frac{1}{n}T(1) + \frac{2}{n}T(2) + \frac{3}{n}T(3) + \cdots \frac{n-1}{n}T(n-1)\right] + n^2$$

$(n+1)\,T(n+1)$

$$nT(n) = 2\left[\frac{1}{n}T(1) + \frac{2}{n}T(2) + \frac{3}{n}T(3) + \cdots \frac{n-1}{n}T(n-1)\right] + n^2$$

$$- \quad n^2 T(n) = 2\left[T(1) + 2T(2) + 3T(3) + \cdots + (n-1)T(n-1)\right] + n^3$$

$$(n+1)^2 T(n+1) = 2\left[T(1) + 2T(2) + \cdots + (n-1)T(n-1) + nT(n)\right] + (n+1)^3$$

$$(n+1)^2 T(n+1) - n^2 T(n) = 2nT(n) + (3n^2 + 3n + 1)$$

$$(n+1)^2 T(n+1) = (n^2 + 2n)T(n) + (3n^2 + 3n + 1)$$

$$\frac{(n+1)^2 T(n+1)}{(n+1)(n+2)} = \frac{n(n+2)T(n)}{(n+1)(n+2)} + \frac{(3n^2 + 3n + 1)}{(n+1)(n+2)}$$

$$\frac{(n+1)T(n+1)}{(n+2)} = \frac{nT(n)}{n+1} + 3$$

Let $S(n) = \frac{nT(n)}{n+1}$

$$S(n+1) = S(n) + 3, \text{ assume } S(0) = 0$$

$$= 3(n+1) = O(n)$$

$S(n) \sim 3n$

$T(n) \sim (1 + \frac{1}{n}) \cdot 3n$

$= 3n + 3$

$$O(n) = \frac{nT(n)}{n+1}$$

$$T(n) = \frac{(n+1) \cdot O(n)}{n} = \left(1 + \frac{1}{n}\right) O(n)$$

$< 2 \cdot O(n) = O(n)$