I & C SCI 46 Winter 2023
Lecture 23: QuickSort

QuickSort Step 1: Partition

89 48

48	41	51	66	84	89	87	68	37	23	96	98	52	14	62	43

1. Choose a pivot. x
2. Place that pivot in the right spot.
3. Pivot the rest of the array.

$< x$ | x | $> x$

sort this and sort this

QuickSort

	48	41	51	66	84	89	87	68	37	23	96	98	52	14	62	43
43	~~89~~		14	23	37	48			84	66				51		89

Running time of QuickSort

- What is the worst-case running time?

  $\Theta(n^2)$

- Can you produce the worst-case running time?

  Min/max as pivot each time

- So why is it called "QuickSort" then?

Selecting a pivot

Variations of QuickSort change based on pivot selection:

- Deterministic, single-spot.

  Adv: easy to code

  disad: worst case easy to generate

- Uniformly at random

- Median-of-3

Average Case Analysis of QuickSort

Suppose

- All permutations equally likely
- All n values are distinct (for simplicity)
- Define $S_1, S_2, \ldots S_n$ as sorted order.

Let $P_{i,j}$ be probability we compare S_i and S_j.

$i<j$

$$P_{ij} = \frac{\#\text{yes}}{\#\text{total}} = \frac{2}{j-i+1}$$

Expected number of comparisons

$$E\Big(\sum_{i=1}^{n}\sum_{j=i+1}^{n} X_{i,j}\Big) = \sum_{i=1}^{n}\sum_{j=i+1}^{n} E(X_{i,j})$$

$$= \sum_{i=1}^{n}\sum_{j=i+1}^{n} \frac{2}{j-i+1}$$

reindex

$$= \sum_{i=1}^{n}\sum_{k=2}^{n-i+1} \frac{2}{k}$$

$$< \sum_{i=1}^{n} \boxed{\sum_{k=1}^{n} \frac{2}{k}}$$

converges ~~sums~~ to $O(\log n)$

indicator random variable : were i,j compared?

is $\Theta(n \log n)$

Hybrid: Tim Sort

- The fastest sort algorithms are recursive
- Problem: recursion has an overhead

- Solution: