Randomization

9-28

QuickSelect

Skip Lists

Treaps

$\boxed{5112}$

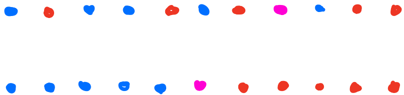
Selection

Given an array with n items
Want to return the k^{th} smallest item.

Quick Select

Select a pivot at random

Partition the array



Random Variables and Expectation

Probability Space

A R.V. associates a number to each event

Given a R.V. X, the expectation of X

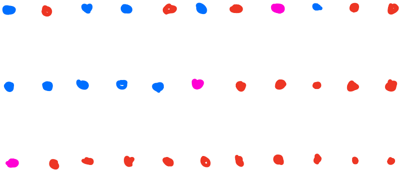
is $E\{X\} = \sum_{i=0}^{\infty} i \Pr\{X = i\}$

Intuitively this is the average outcome of X.

What's the expected running time of QuickSelect?

Define $T\{n,k\}$ to be the expected running time of k-QuickSelect.

Define $T\{n\} = \max\limits_{1 \le k \le n} T\{n,k\}$

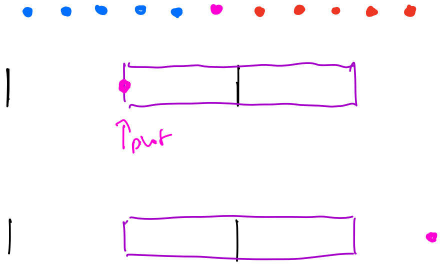


bad case

the recursion goes to $T\{n-1\}$

Define $T\{n,k\}$ to be the expected running time of k-QuickSelect.

Define $T\{n\} = \max\limits_{1 \le k \le n} T\{n,k\}$



"good pivot"

Problem size reduces to $\le \frac{3}{4}n$

"bad pivot"

Problem size is basically "the same".

$\frac{1}{2}$ good, $\frac{1}{2}$ bad

$$T\{n\} = \underset{\substack{\uparrow \\ \text{time to partition}}}{n} + \frac{1}{2} T\{\tfrac{3}{4}n\} + \frac{1}{2} T\{n\}$$

$$\frac{1}{2} \cdot T\{n\} = n + \frac{1}{2} T\{\tfrac{3}{4}n\} \implies T\{n\} = O(n)$$

So the expected runtime is $O(n)$.

14	14
18	
23	
28	
34	34
42	42
50	
59	
66	
72	72

A sorted linked list

$14 \to 18 \to 23 \to 28 \to 34 \to 42 \to 50$

A really bad search data structure

$O(n)$ insert

$O(n)$ search

14	14
18	
23	
28	
34	34
42	42
50	
59	
66	
72	72

Find 50

14 → 34 → 42 → 72

14 → 18 → 23 → 28 → 34 → 42 → 50 → 72

14 34

14 23 34

14 → 18 → 23 → 28 → 34

Insert 20

If the pattern is fixed, expensive to insert.

How to up/down-grade many items

Skip list

Have 2log n levels to store n items,

Each level is a sorted linked list.

When an item is inserted, flip a coin to determine whether it get upgraded, if so, repeat.

What is the expected cost of a search?

Head ⟶ 28 ⟶ 72

Head ⟶ 14 ⟶ 28 ⟶ 50 ⟶ 72

Head ⟶ 14 ⟶ 23 ⟶ 28 ⟶ 34 ⟶ 42 ⟶ 50 ⟶ 66 ⟶ 72

How many steps did we take on level 1?

$1 \quad \frac{1}{2}$ time take a step

$1 \quad \frac{1}{4}$ time take a 2nd step

$1 \quad \frac{1}{8} \quad \cdot \quad \cdot \quad \cdot$

$$= \sum_{i=1}^{\infty} \frac{1}{2^i} \leq 1.$$

$$E\{\#\text{ of steps}\} \leq \sum_{i=1}^{\infty} \Pr\{i \text{ steps}\} \cdot i$$

$$= \sum_{i=1}^{\infty} \frac{i}{2^i} \leq 2.$$

$\Rightarrow E\{\#\text{ of steps on level } k\} \leq 2.$ for each k.

2 expected steps per level $\times$ 2 log n levels,

$$E\{\#\text{ of steps}\} = O(\log n).$$

Treap / RBST

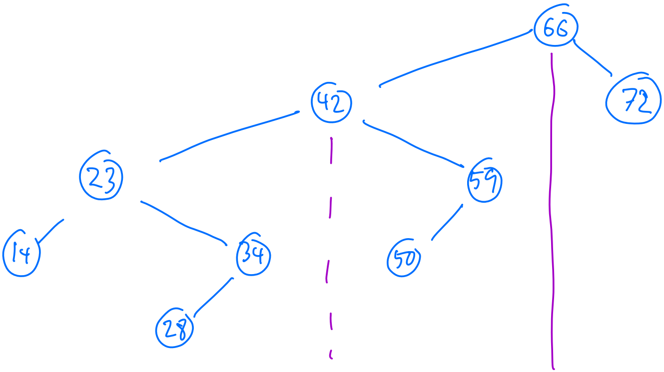
Associate a "priority" to each item randomly.
↳ a hash value on $\{0, 1\}$.

Then a treap is a BST which satisfies the property that every parent node has lower priority than its children.

Heap Property.

BST Property

Each parent has larger key than its left child and smaller than its right child.



Key	14	23	28	34	42	50	59	66	72
Priority	81	26	95	51	15	70	66	3	58