15-750
1/21/09

Binary Search Trees

Data Structure Dictionary

S is an ordered set.

1) Search$(K, S) \equiv K \in S$?
2) Insert$(K, S) \equiv$
3) Delete(K, S)

Note If 1), 2), 3) are the Design Requirement then use a Hash table

4) Range$(K, K', S) \equiv \{K'' \in S \mid K \leq K'' \leq K'\}$

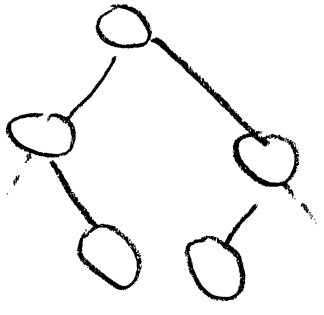
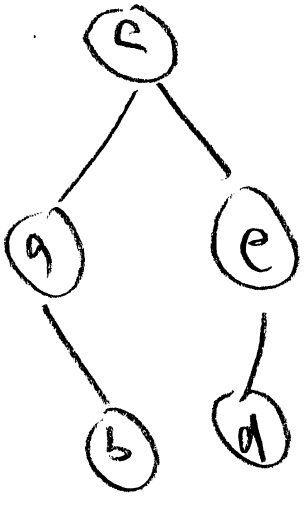
If 1), ..., 4) use BST

<u>Def</u> A tree T is BST for keys S if:

1) T is an ordered binary tree with $|S|$ nodes.
2) Each node stores a key.
3) Keys are in inorder

---

eg $S = \{a, b, c, d, e\}$ BST

$T =$  $\Rightarrow$ 

T is <u>balanced</u> if $\text{Math depth}(T) = O(\log n)$

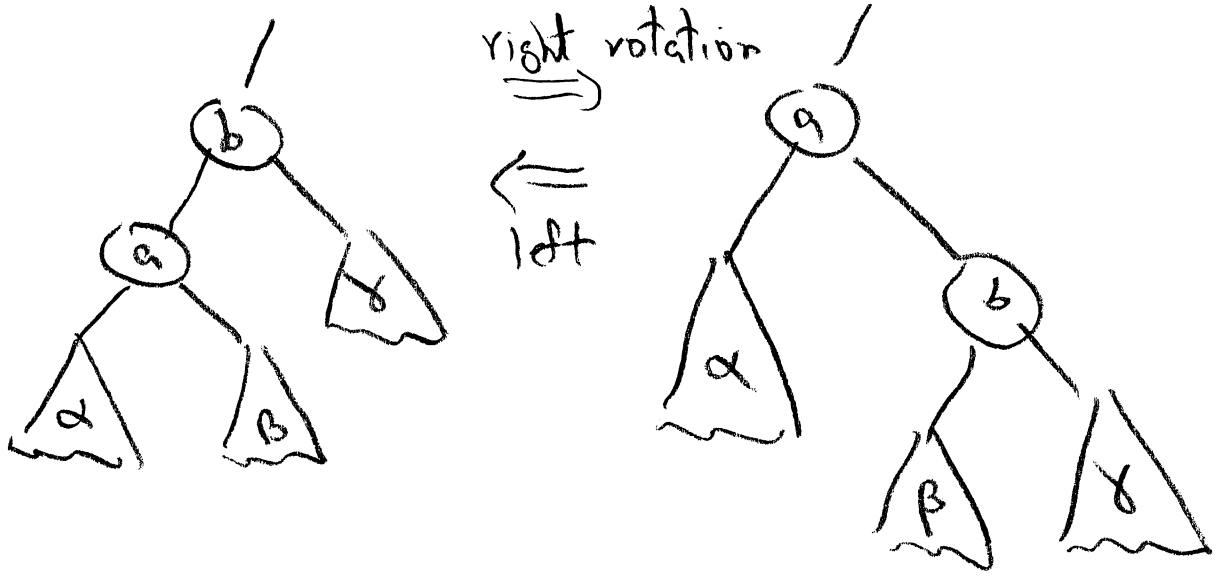
Types of Balanced BST's

Always Balanced - AVL, 2-3-4, RB, B Trees

Randomized - skip-lists, Treaps = tree-heaps

Amortized - Splay Trees

All these use the **Rotation**

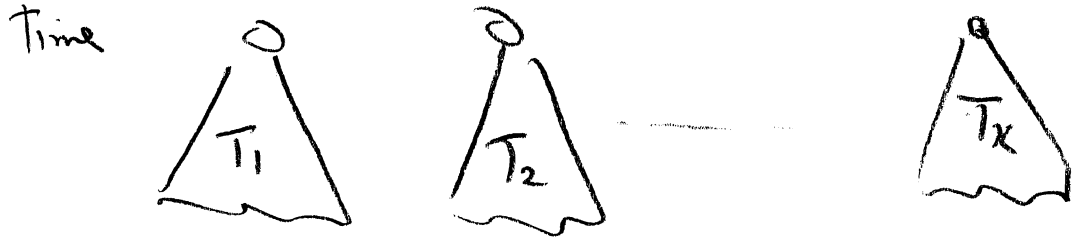


To show: inorder is preserved

Applications

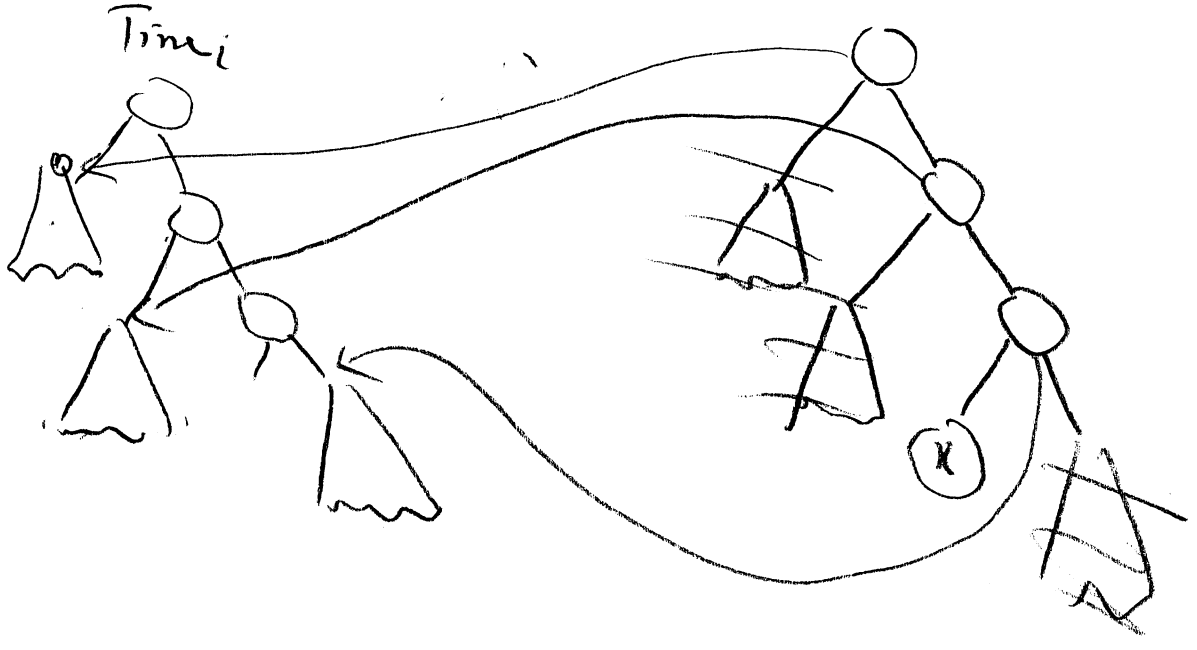
Persistance (undo)

First Idea Keep a tree for each time



$O(n^2)$ spaces.

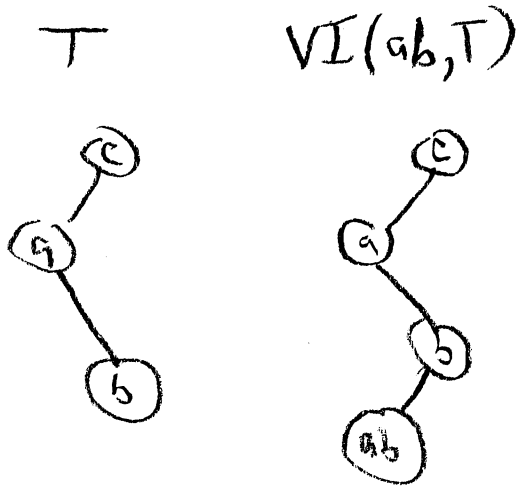
Idea 2 store only diffences Insert(x)



$O(\log n)$ space per new tree.

Vanilla-Insert (K T)

1) Find empty leaf node
2) add K to leaf



QS(A) Bad Quick Sort

1) Pick first a in A
2) Split S into S<a, a, a<L
3) Return QS(S), a, QS(L)

Eager

Vanilla-BST(A,T) VBST

Lazy

1) Extract first a from A
2) Return VBST(A, VI(a,T))

<u>Note</u> QS & V-BST do exactly the same comparisons but in different order.

Treaps

Keys $\equiv 1, \cdots, n$

Priorities $\equiv p(k)$ $\quad p(k) \neq p(l)$ for $k \neq l$

$T \equiv$ tree with a key at each node.

<u>Def</u> T is in <u>heap order</u> if $\forall x \in T$ x not root

$$p(\text{parent}(x)) < p(x)$$

<u>Lemma</u> A heap order BST exists and is unique.

<u>pf</u> Vanilla-insert in priority order.

Random Treap

Insert(K) ≡ 1) insert K into a leaf
2) pick random p(K)
3) rotate K up until in heap order

Delete(K) ≡ 1) Rotate K to a leaf
by picking highest priority child
2) remove K.

Correctness:

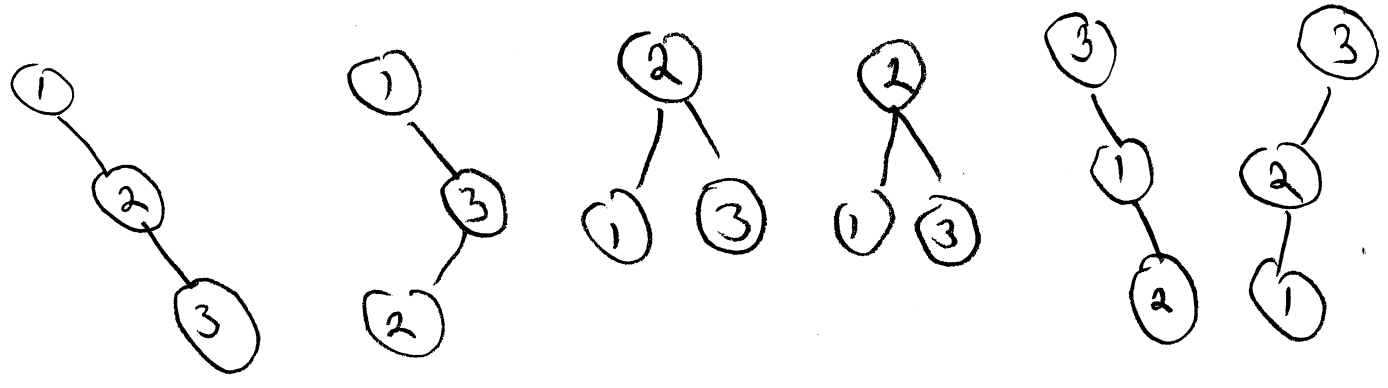
Expected Cost for Treaps

Goal: Determine expected # of comparisons to search(m.5, K) $\equiv S(m.5)$

eg $K = \{1,2,3\}$ & search $(2.5, K)$

Treaps insert order

(123) (132) (213) (231) (312) (321)



#	3	3	2	2	3	2

$$S(2.5) = \frac{15}{6} = 2\tfrac{1}{2}$$

Note for random BST $\frac{13}{5} = 2\tfrac{3}{5}$

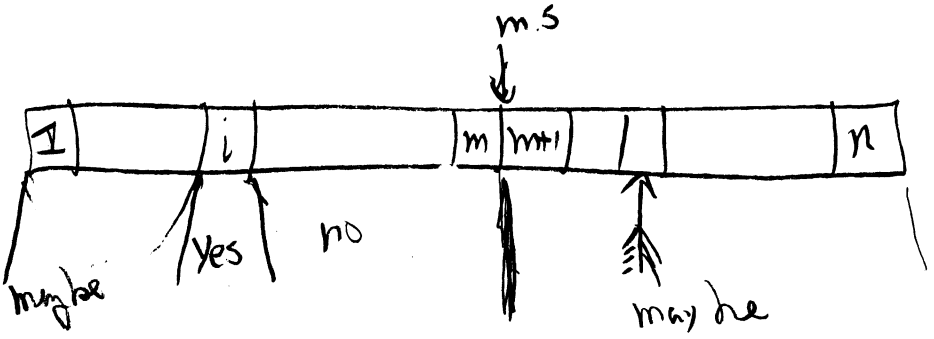
Keys $\{1, \ldots, n\}$ in Treap

$C(i,m)$ = Event [i is compared to m in Search(m)]

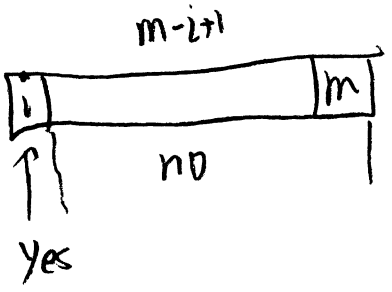
<u>Claim</u> $\text{Prob}[C(i,m.5)] = \dfrac{1}{m-i+1}$ if $i \le m$

$= \dfrac{1}{i-m}$ if $i > m$

Treap construction at dart game



informal argument: WLOG board is



$\text{Prob}[C(i,m.5)] = \dfrac{1}{m-i+1}$

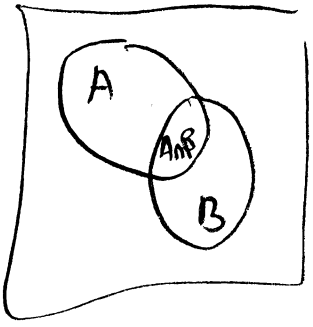
Formal Arugment

B_j = Event [jth dart is first to land in $[i, \ldots, m]$]

$B_k \cap B_j = \emptyset$ if $k \neq j$ $\quad Pr\left(\bigcup_{j=1}^{\infty} B_j\right) = 1 \qquad (*)$

Conditional Probability: Let A, B be events

$$Pr[A|B] = \frac{Pr[A \cup B]}{Pr[B]}$$



$i \leq m$

$$Pr[\underbrace{C(i,m,\xi)}_{=A}] = \sum_{j=1}^{\infty} Pr[B_j]\, Pr[A|B_j] \quad \text{from by } (*)$$

<u>note</u> $Pr[A|B_j] = \frac{1}{m-i+1}$

$$Pr[A] = \sum_j Pr[B_j]\left(\frac{1}{m-i+1}\right) = \frac{1}{m-i+1}\sum_{j=1}^{\infty} Pr[B_j] = \frac{1}{m-i+1}$$

$S(m.5)$ = # comparison search $m.5$

$$= \sum_{i \neq m} C(i, m.5)$$

$$E(S(m.5)) = \sum E(C(i, m.5)) = \{Prob[C(i, m.5)]\}$$

$$= \sum_{i \leq m} \frac{1}{m-i+1} + \sum_{i > m}^{n} \frac{1}{i-m} \leq 2 \sum_{i=1}^{n} \frac{1}{i} = 2 H_n$$

$$= 2 \ln n + O(1)$$

Counting Rotations

2 Cases insert, delete

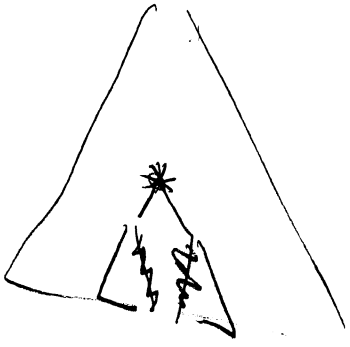
Delete: Move-to-Leaf

Claim Expect # of rotations < 2

Claim Possible pivots with x

1) right-most nodes in left subtree of x

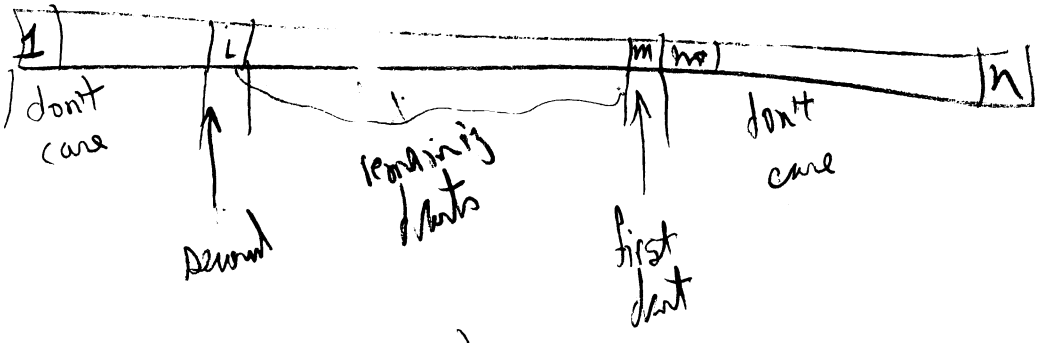
2) left " " right " "



D_i = Event [i is a right most node in left subtree of m]

left right

Dart Game Informal $Pr[D_i]$



$$Pr[D_i] = \left(\frac{1}{m-i+1}\right)\left(\frac{1}{m-i}\right)$$

R_m = # rotations move-to-leaf of m

$$E(R_m) \le \sum_{i \ne m} E(D_i)$$

$$\le \sum \frac{1}{(m-i+1)(m-i)} + \sum \frac{1}{(i-m-1)(i-m)}$$

$$\le \sum_{i=1}^{m-1} \frac{1}{(i+1)\,i} + \sum_{i=1}^{n-m} \frac{1}{(i+1)\,i}$$

$$\left(\frac{1}{(i+1)\,i} = \frac{1}{i} - \frac{1}{i+1} \right) \qquad \sum_{i=1}^{m-1} \frac{1}{i} - \sum_{i=2}^{m} \frac{1}{i}$$

$$= 1 - \frac{1}{m} < 1$$

$$< 2$$