Reminders:

```
FindMax(a, n).
  max ← -∞
  for i = 1 to n:
    if a_i > max then
      max ← a_i   (*)
  return max.
```

Theorem: If a is a random permutation of $\{1 \dots n\}$ then the expected number of times line (*) executes is $H_n \stackrel{def}{=} \sum_{i=1}^{n} 1/i$

n^{th} harmonic number.

$$\ln n \le H_n \le 1 + \ln n.$$

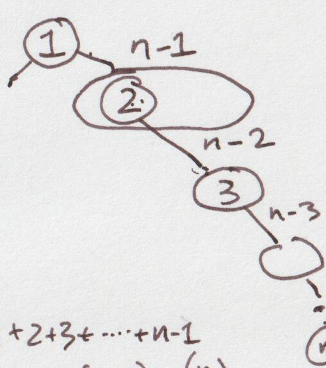
```
def quicksort(a):
    if len(a) ≤ 1:
        return a
    p = random.choice(a)
    return quicksort([x for x in a if x<p]) \
        + [x for x in a if x==p] \
        + quicksort([x for x in a if x>p]).
```

a: 2 5 1 7 6 8 3 4

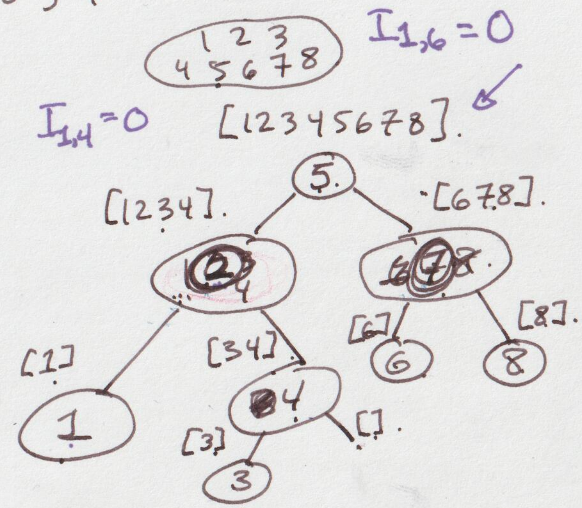
$I_{2,5} = 1$

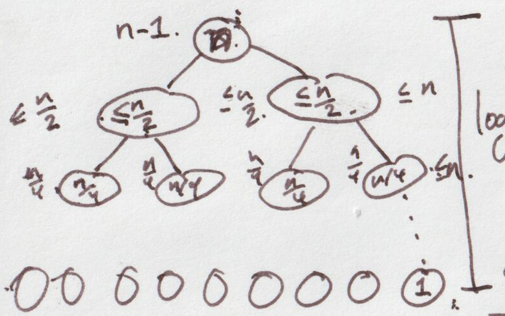
$I_{1,6} = 0$

$I_{1,4} = 0$



$1+2+3+\dots+n-1$
$= \frac{n(n-1)}{2} = \binom{n}{2}$





Quicksort

Best case	Expected	Worst case
$\approx n \log_2 n$	$2n \cdot \ln n$	$\binom{n}{2}$

InsertionSort (# comparison)

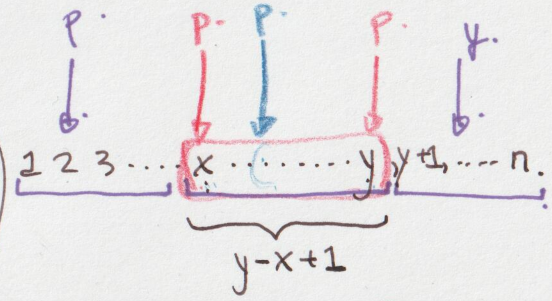
Best case	Expected	Worst case
$n-1$	$\binom{n}{2}/2$	$\binom{n}{2} = \frac{n(n-1)}{2}$

Let X be the number of comparisons performed by quicksort when sorting $a = [1, 2, 3, 4, \ldots, n]$.

For each x, y with $1 \le x < y \le n$ define $I_{x,y} = \begin{cases} 1 & \text{if quicksort compares } x \text{ and } y. \\ 0. & \text{otherwise.} \end{cases}$

$$E(X) = E\left(\sum_{x=1}^{n-1}\sum_{y=x+1}^{n} I_{x,y}\right) = \sum_{x=1}^{n-1}\sum_{y=x+1}^{n} E(I_{x,y}) = \sum_{x=1}^{n-1}\sum_{y=x+1}^{n} Pr(I_{x,y} = 1)$$

$$= \sum_{x=1}^{n-1}\sum_{y=x+1}^{n} \frac{2}{y-x+1} = 2\sum_{x=1}^{n-1}\left(\frac{1}{(x+1)-x+1} + \frac{1}{(x+2)-x+1} + \frac{1}{(x+3)-x+1} + \cdots + \frac{1}{n-x+1}\right)$$



$$= 2\sum_{x=1}^{n-1}\left(\underbrace{\frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \cdots + \frac{1}{n-x+1}}_{H_{n-x+1}-1}\right) = 2\sum_{x=1}^{n-1}\left(H_{n-x+1} - 1\right)$$

$$Pr(I_{x,y} = 1) = \frac{2}{y-x+1}.$$

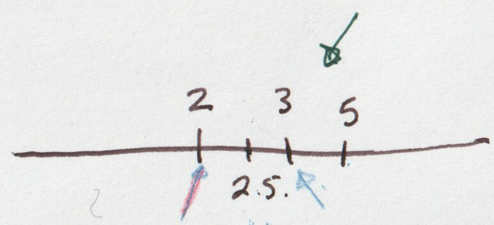
$$\le 2\sum_{x=1}^{n}(H_n - 1) = 2n(H_n - 1) \le 2n \cdot \ln n.$$

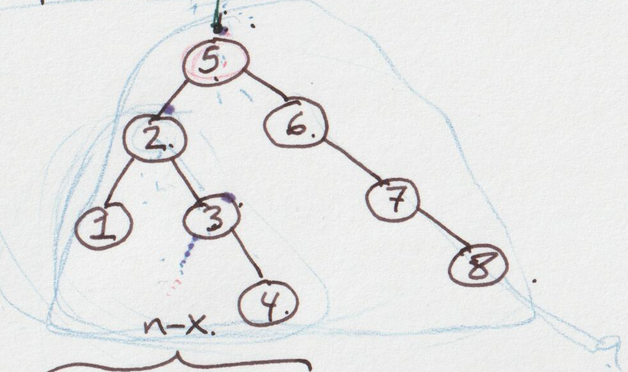
<u>Def'n</u>: A <u>random binary search tree</u> (RBST) is a binary search tree (BST) made by inserting a random permutation of $\{1,\dots,n\}$ into an initially empty binary search tree.

$\pi = 5\ 2\ 6\ 7\ 3\ 1\ 4\ 8$

$x+\frac{1}{2}$ for any $x \in \{0,\dots,n\}$

2.5





C_x = the ~~expected~~ # of comparisons when searching for $x+\frac{1}{2}$.

$C_{2.5} = 3$.

$\pi_{>x} = \overbrace{5\ 6\ 7\ 3\ 4\ 8}^{n-x}$

$\pi_{<2.5} = \underbrace{2\ 1}_{x}$

$$E(C_x) = H_x + H_{n-x}$$

$$\le 2 + 2\ln n.$$