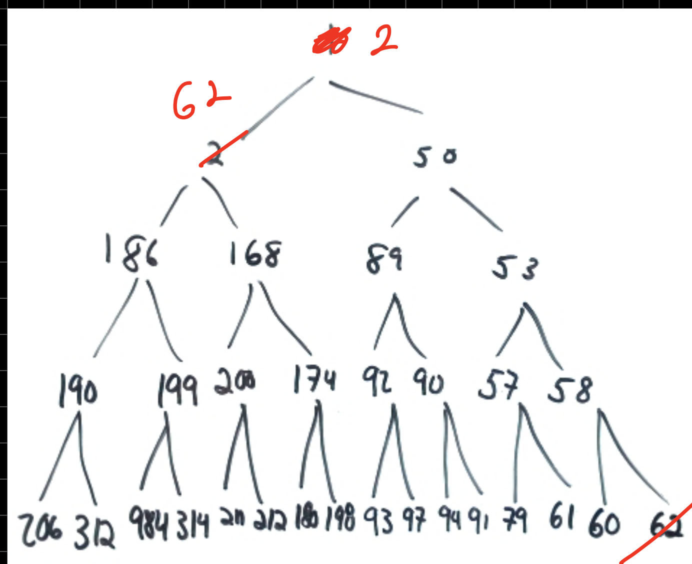
• 2d array's in java are initalized to $O(n^2)$ time



must be $O(\log n \cdot 1) = O(\log n)$
since getting the last value inserted is $O(h)$ $= O(\log n)$ & percolate down costs $O(1)$ (in this instance) therefore $O(\log n)$ is the cost of time to delete min which is root which is 1

Big Oh, Theta, Omega

$f(n) = O(g(n))$ iff $f(n) \leq c_1 \cdot g(n)$ for $n \geq N_0$

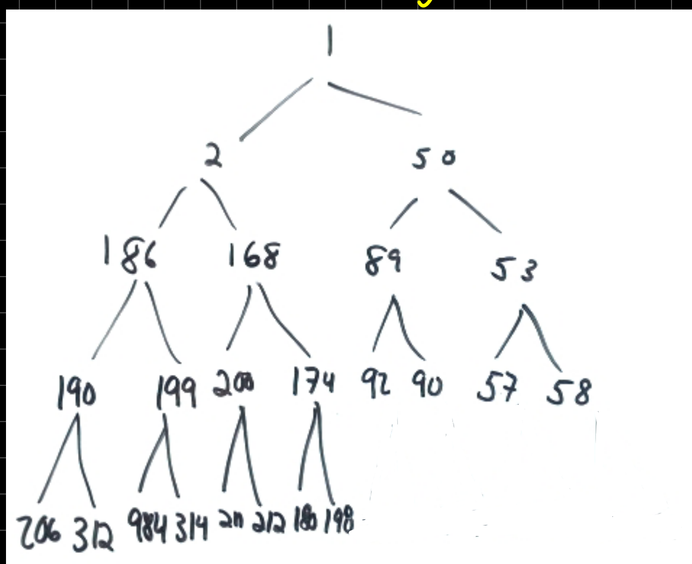
N_0 means the point in which $c_1 g(n) \geq f(n)$
c_1 is the constant in front of n in $f(n)$

1) 1. $2n^2 + 6 = f(n)$ $\qquad g(n) = n^2$

$f(n) \leq c_1 \cdot g(n)$ $\qquad c_1 = 2.2$

$1.2n^2 + 6 \leq$

another thought



- how do you know the last node?
  - need either size of last node in order to make this a $O(1)$ deletion

X because it should be a fucking array not a heap

also java stores the length

Attempt 2 at asymptotic Bounding (Big Oh, Big theta, Big Omega)

- Big Oh $O(\ )$ is the upper bound

- $T(n)$ is $O(f(n))$ iff $T(n) \leq c f(n)$ for all $n \geq n_0$

new definition for n_0 is initial value

Redo:

1) $1.2n^2 + 6 = f(n)$

Finding Big Oh (tight upper bound) where $g(n)$ is the smallest function such that $f(n)$ is $O(g(n))$

$1.2n^2 + 6 \leq 1.2n^2 + 6n^2$ such that $\underbrace{n \geq 1}_{\text{no function runs less than 1 time when called}} = N_0$

$1.2n^2 + 6 \leq 7.2n^2$

$O(g(n)) = O(n^2)$ $c = 7.2$

Runtime Analysis Quiz

$T_1(n) = \frac{1}{6}n^3 + 1000n^2$

$\frac{n(n+1)}{2}$

$\sum_{i=1}^{n-1} 5 + i$

$= \sum_{i=1}^{n-1} 5 + \sum_{i=1}^{n-1} i$

$= 5(n-1) + \left[\frac{(n-1)(n)}{2}\right]$

say $n = 3$

$i = 1$ $1 < 3$ $i{+}{+}$

$x += 5 + 1 = 6$

$i = 2$ $2 < 3$ $2{+}{+}$

$x += 5 + 2 = 7 + 6 = 13$

$n=3$

$= 5(2) + \left[\frac{(2)(3)}{2}\right]$ $i=3$ ~~3 < 3~~

$= 10 + 3 = 13$

$n=3$

$5 \sum_{i=1}^{n-1} 1 + i \sum_{i=1}^{n-1} 1$

$5(2) + i(2) = 10 + 2i$

$1.2n^2 + 6 \leq 2.2n^2$

$n^2 \geq 6$ $C = 2.2$

$n \geq \sqrt{6}$ for $n \geq \sqrt{6}$

therefore n^2 is Big Oh since the graph multiplied by the constant 2.2 will always be larger then it

$1.2n^2 + 6 \geq 6$ for all $n \geq 1$

$1.2n^2 + 6 \geq$

$1.2n^2 + 6 \leq n + 10$ if $c_1 = 0$

$1.2n^2 + 6 \leq 10$ ← then

allowing proof that $n^2 \leq n$ for an upper bound

1) $f(n) = \frac{1}{8}n^2 + \frac{1}{8}$

prove $O(n^2)$

$$\frac{1}{8}n^2 + \frac{1}{8} \le \frac{1}{8}n^2 + n^2$$

$$\frac{1}{8}n^2 + \frac{1}{8} \le \frac{9}{8}n^2 \qquad \text{for } n \ge \underbrace{\sqrt{\tfrac{1}{8}}}^{N_0}$$

$$\frac{1}{8} \le n^2 \qquad c = \frac{9}{8}$$

$$g(n) = n^2$$

prove $o(n^3)$

$$\frac{1}{8}n^2 + \frac{1}{8} \le n^3 + \frac{1}{8}n^2$$

$$\frac{1}{8} \le n^3 \qquad \text{for } n \ge \sqrt[3]{\frac{1}{8}} = \frac{1}{2}$$

prove $\Omega(n^2)$

$$\frac{1}{8}n^2 + \frac{1}{8} \ge \frac{1}{8}n^2$$

$$\frac{1}{8} \ge 0 \qquad \text{for all } n$$

prove $\Omega(n)$

$$\frac{1}{8}n^2 + \frac{1}{8} \ge \frac{1}{8}n \ge \frac{1}{8} \qquad \text{for } n \ge 1$$

$$\lim_{n\to\infty} \frac{\frac{n^2}{8}}{\frac{n}{8}} = \lim_{n\to\infty} \frac{n^2}{8} \cdot \frac{8}{n} = \lim_{n\to\infty} n$$

$$= \infty$$

$\frac{1}{8}n^2 + \frac{1}{8} \geq Cn$

AA - ZZ

$c_1 c_2$

Sean AAAA → ZZZZ

$\underbrace{26 \times 26 \times 26 \times 26}$

2x2 possibilities

00
01
10
00

2^2

c_1 — 26 possibilities

26^4 possibilities for AAAA → ZZZZ

from 100000 strings is

$\frac{100000}{26^4}$ = 22% chance

~~not right because you can have duplicates~~

$\frac{1}{26^4}$ + itself 100000 times still right

$\frac{5(100000)}{26^4}$

$\sum_{i=1}^{100000} \frac{1}{26^4}$

$\sum_{i=1}^{100000} \frac{5}{26^4}$

$$\left(\frac{26^4-1}{26^4}\right)-1$$