Xueying Chen

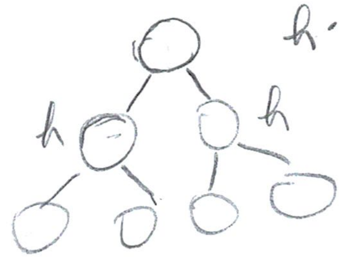
DSA 4413 - Midterm #1

1) a)

$i = \lfloor \frac{Len(A)}{2} \rfloor$

for i down to 1

max-heapify(A, i)



Max-heapify has max of h operations where h = height of the tree

$$T(n) = \sum_{h=1}^{\log(n)} n \cdot \frac{h}{2^h} = n \cdot \sum_{h=1}^{\log(n)} \frac{h}{2^h}$$

$$\sum_{h=1}^{\log(n)} \frac{h}{2^h} = \frac{1}{2} + \frac{2}{2^2} + \frac{3}{2^3} + \frac{4}{2^4}$$

$$= \frac{1}{2} + \frac{2}{4} + \frac{3}{8} + \frac{4}{16}$$

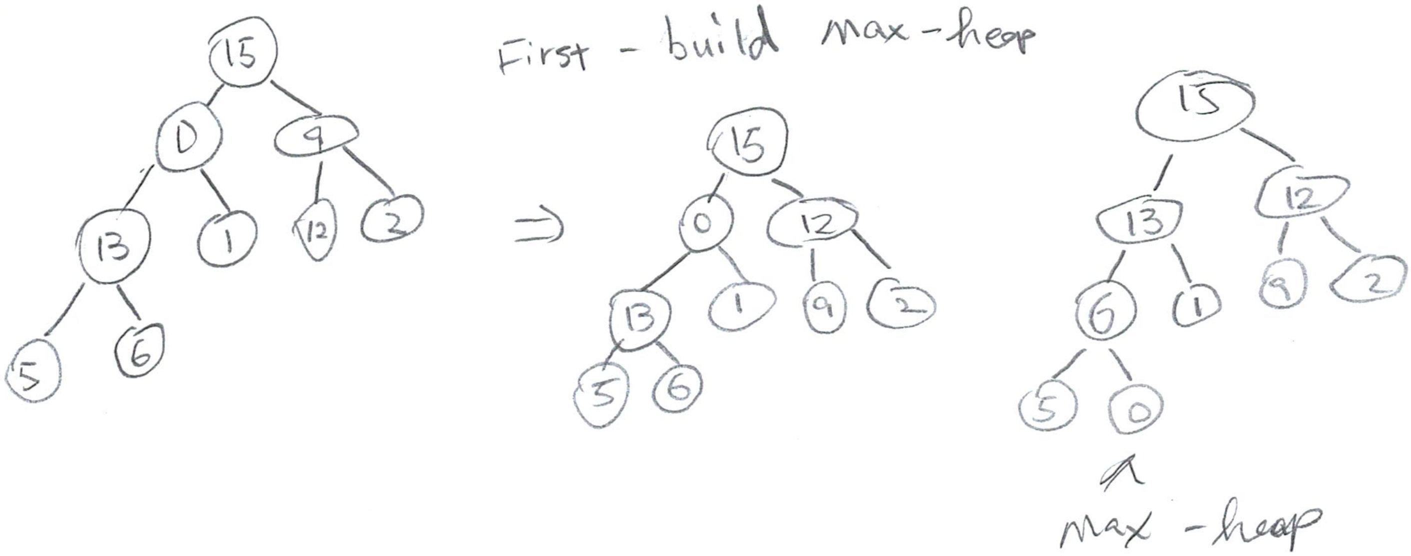
this is geometric series with $x < 1$

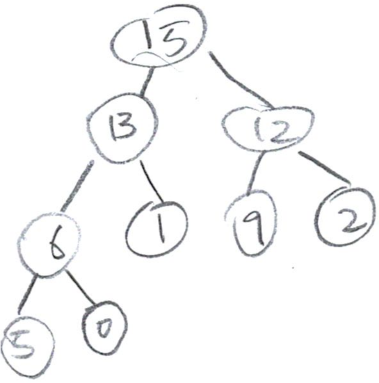
$$\sum_{h=1}^{\log(n)} \frac{h}{2^h} = c \text{ constant} \leq 2$$

$$T(n) = cn = O(n).$$

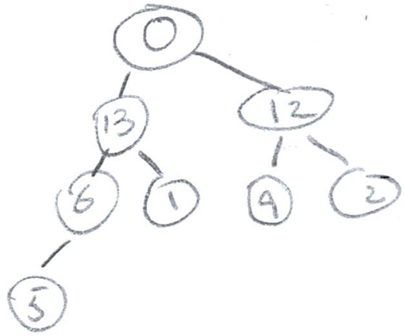
b)

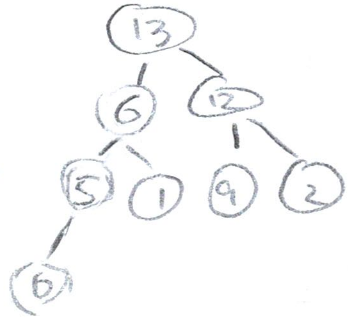
First - build max-heap



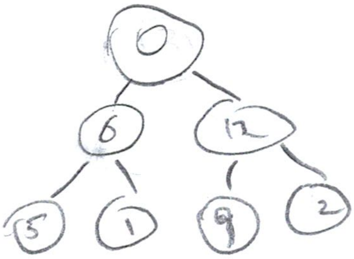


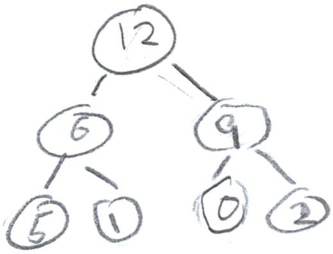
15



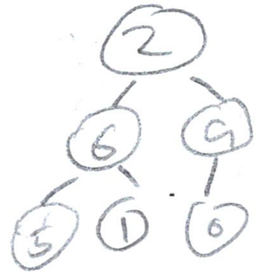


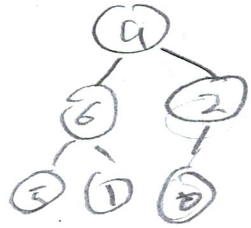
13



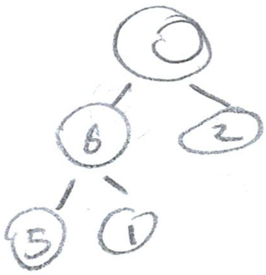


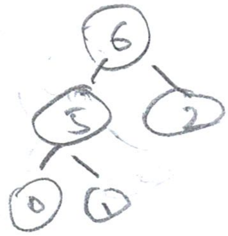
12



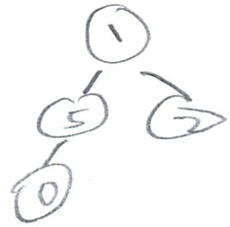


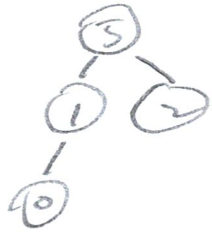
9



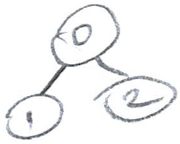


6





5



2

1

0

15, 13, 12, 9, 6, 5, 2, 1, 0

c) worst-cast for Quick-Sort when pivot is the largest or smallest element in every iteration

$$T(n) = \underset{\text{worst-case scenario.}}{T(n-1)} + \underset{\text{number of operation}}{n-1}$$

$$T(n) = O(n^2)$$

(2)

a) $T(n) = 3T(\frac{n}{2}) + n$, $T(1) = 1$ $n = 2^k$

$$T(\tfrac{n}{2}) = 3T(\tfrac{n}{2^2}) + \tfrac{n}{2}$$

$$T(n) = 3(3T(\tfrac{n}{2^2}) + \tfrac{n}{2}) + \tfrac{n}{1}$$

$$T(n) = 3^2 T(\tfrac{n}{2^2}) + 3(\tfrac{n}{2}) + \tfrac{n}{1}$$

$$T(\tfrac{n}{2^2}) = 3T(\tfrac{n}{2^3}) + \tfrac{n}{2^2}$$

$$T(n) = 3^2(3T(\tfrac{n}{2^3}) + \tfrac{n}{2^2}) + 3(\tfrac{n}{2}) + \tfrac{n}{1}$$

$$T(n) = 3^3 T(\tfrac{n}{2^3}) + \frac{3^2 n}{2^2} + 3(\tfrac{n}{2}) + \frac{3^0 n}{2^0}$$

$$T(n) = 3^k T(\tfrac{n}{2^k}) + n\sum_{i=0}^{k-1}(1.5)^i$$

$n = 2^k$

$$\frac{n}{2^k} = 1 \Rightarrow k = \log n$$

$$T(n) = 3^{\log n} + n\sum_{i=0}^{k-1}(1.5)^i \Rightarrow \frac{(1.5)^k - 1}{1.5 - 1} = 2(1.5^k - 1)$$

$$T(n) = 3^{\log n} + 2n(1.5^{\log n} - 1)$$

$$T(n) = n^{\log 3} + 2n(n^{\log(1.5)} - 1)$$

$$\boxed{T(n) = n^{0.48} + 2n(n^{0.18} - 1)}$$

b) $T(n) = 2T\left(\frac{n}{2}\right) + n \quad T(1) = 1, \quad n = 2^k$

$$T\left(\frac{n}{2}\right) = 2T\left(\frac{n}{2^2}\right) + \frac{n}{2}$$

$$T(n) = 2\left(2T\left(\frac{n}{2^2}\right) + \frac{n}{2}\right) + n$$

$$T(n) = 2^2 T\left(\frac{n}{2^2}\right) + \frac{2n}{2} + n$$

$$T\left(\frac{n}{2^2}\right) = 2T\left(\frac{n}{2^3}\right) + \frac{n}{2^2}$$

$$T(n) = 2^2\left(2T\left(\frac{n}{2^3}\right) + \frac{n}{2^2}\right) + \frac{2n}{2} + n$$

$$T(n) = 2^3 T\left(\frac{n}{2^3}\right) + \frac{\cancel{2^2} n}{\cancel{2^2}} + \frac{\cancel{2} n}{\cancel{2}} + n$$

$$T(n) = 2^k T\left(\frac{n}{2^k}\right) + kn$$

$n = 2^k \qquad \frac{n}{2^k} = 1 \Rightarrow \log n = k$

$$T(n) = \cancel{2}^{\log n} + \log n \; n$$

$$T(n) = n + n \log n$$

$$T(n) = n \log n + n$$

③

A1	A2	A3	A4
15x5	5x10	10x20	20x25

	1	2	3	4
1	X	750	2500	5375
2		X	1000	3500
3			X	5000
4				X

	1	2	3	4
1	X		1	1
2		X		4
3			X	
4				X

$M_{12} = 15 \times 5 \times 10 = 750$

$M_{23} = 5 \times 10 \times 20 = 1000$

$M_{34} = 10 \times 20 \times 25 = 5000$

$$M_{13} = \min \begin{pmatrix} M_{11} \cdot M_{23} + C(M_{23}) \checkmark \\ M_{12} \cdot M_{33} + C(M_{12}) \end{pmatrix}$$

$$= \min (15 \times 5 \times 20 + 1000, \ 15 \times 10 \times 20 + 750)$$

$$= 2500$$

$$M_{24} = \min \begin{pmatrix} M_{22} \cdot M_{34} + C(M_{34}) \\ M_{23} \cdot M_{44} + C(M_{23}) \end{pmatrix} \checkmark$$

$$\min (5 \times 10 \times 25 + 5000, \ 5 \times 20 \times 25 + 1000) = 3500$$

$$M_{14} = \min \begin{pmatrix} M_{11} \cdot M_{24} + C(M_{24}) \\ M_{12} \cdot M_{3\cdot4} + C(M_{12})\, C(M_{34}) \\ M_{1\cdot3} \cdot M_{44} + C(M_{13}) \end{pmatrix}$$

$$= \min \begin{pmatrix} 15 \times 5 \times 25 + 3500, \\ 15 \times 10 \times 25 + 750 + 5000, \\ 15 \times 20 \times 25 + 2500, \end{pmatrix}$$

$$= 5375$$

$$(A_1)((A_2 A_3)(A_4))$$

optimal parenthesis

(4) inversion of permutation is

(1) for $a_i\ a_j \in A$
if $i<j$ then $a_i > a_j$

(2) the number of swap we have to do during Bubble sort is equal to number of inversion pair we have in A.

(3) $(8,6)$, $(8,1)$ $(6,1)$
$(2,1)$ $(3,1)$

running time of bubble sort is porportional to number of inversion pair.
if there n^2 inversion pair then
$T(n) = O(n^2)$