Algorithms -- COMP.4040 Honor Statement
(Courtesy of Prof. Tom Costello and Karen Daniels with modifications)

Must be attached to each submission

Academic achievement is ordinarily evaluated on the basis of work that a student produces independently. Infringement of this Code of Honor entails penalties ranging from reprimand to suspension, dismissal or expulsion from the University.

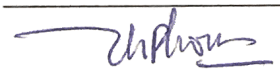
Your name on any exercise is regarded as assurance and certification that what you are submitting for that exercise is the result of your own thoughts and study. Where collaboration is authorized, you should state very clearly which parts of any assignment were performed with collaboration and name your collaborators.

In writing examinations and quizzes, you are expected and required to respond entirely on the basis of your own memory and capacity, without any assistance whatsoever except such as what is specifically authorized by the instructor.

I certify that the work submitted with this assignment is mine and was generated in a manner consistent with this document, the course academic policy on the course website on Blackboard, and the UMass Lowell academic code.

Date: 6/16/2019

Name (please print): PHONG VO

Signature: 

Due Date: 06-17-2019 (M), BEFORE the class begins

This assignment covers textbook Chapter 6 and Chapter1~5.

1. Heaps (15 points)
Given the set of integers: {88, 2, 9, 36}, how many different MIN HEAPs can be made using these integers? Justify your answer.

2. **Heap property** (15 points)
Textbook Exercise 6.2-6 (p156)

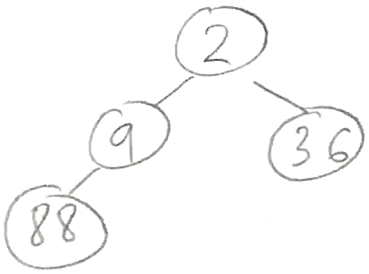
3. **Heap and Heap property** (15 points)
Provide a tight bound for the running time of finding the biggest element in a binary min-heap with n elements? Justify your answer.

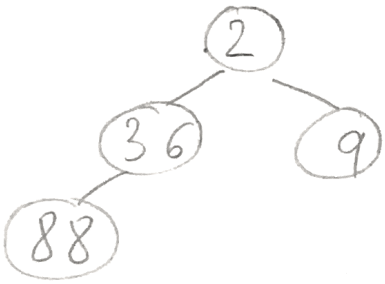
4. **Heap Sort** (15 points)
Using Figure 6.4 as a model, illustrate the operation of HEAPSORT on the array
A = <5, 12, 1, 55, 8, 17, 22, 9, 4>

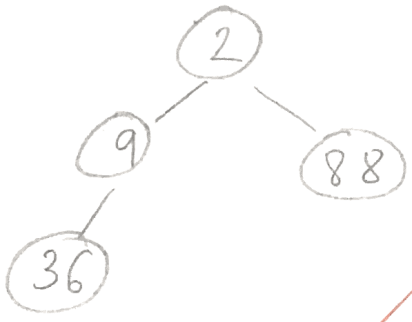
5. **Priority Queue** (40 points)
Textbook Exercise 6.5-8 (P 166). Provide (1) pseudocode, (2) correctness justification, and (3) provide an upper bound of your procedure and give an explanation.

HW6

1/ Given set {88, 2, 9, 36}







2/

Show that the worst-case running time of MAX-HEAPIFY on a heap of size n is Ω(lg n). (Hint: For a heap with n nodes, give node values that cause MAX-HEAPIFY to be called recursively at every node on a simple path from the root down to a leaf.)

Answer:

The worst-case occurs when the A[root] is the least value in a max-heap A. Since then, A[i] needs to recursively traverses on a longest path from top down-to it becomes a leaf. In addition, the longest path is a continuous connection of h levels (whereas h = *lg* n is the height of the heap). So, the function MAX-HEAPIFY is needed to call *lg* n times, which is ~~Θ~~(*lg* n). Hence, the worst-case running time is Ω(*lg* n)

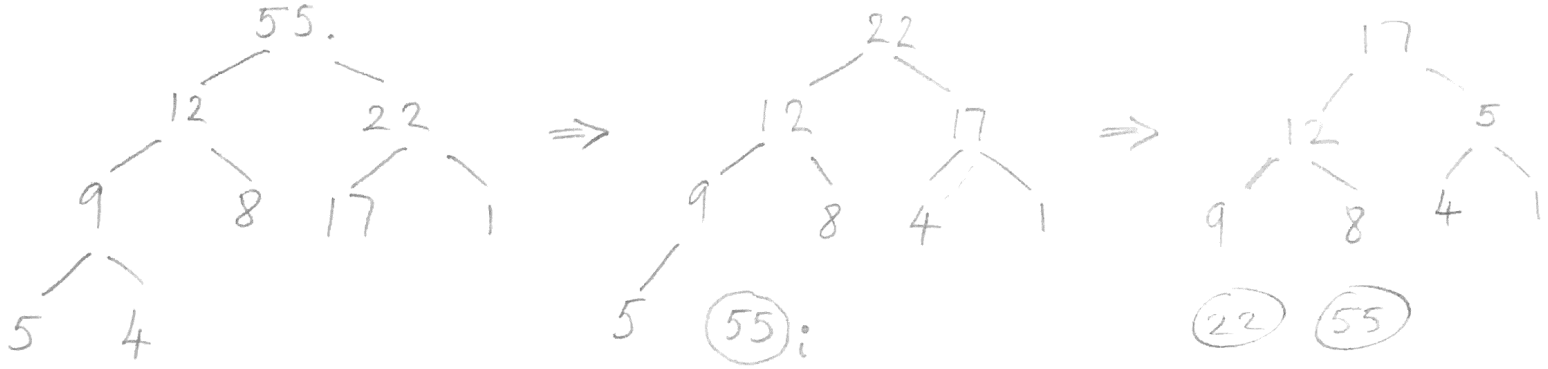
3/

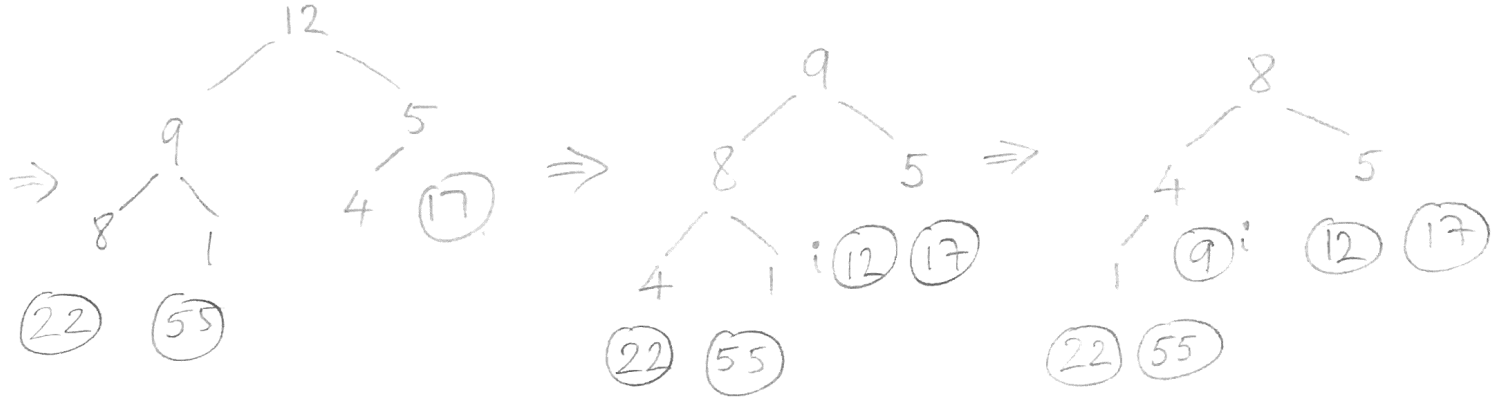
Provide a tight bound for the running time of finding the **biggest** element in a binary **min-heap** with n elements.

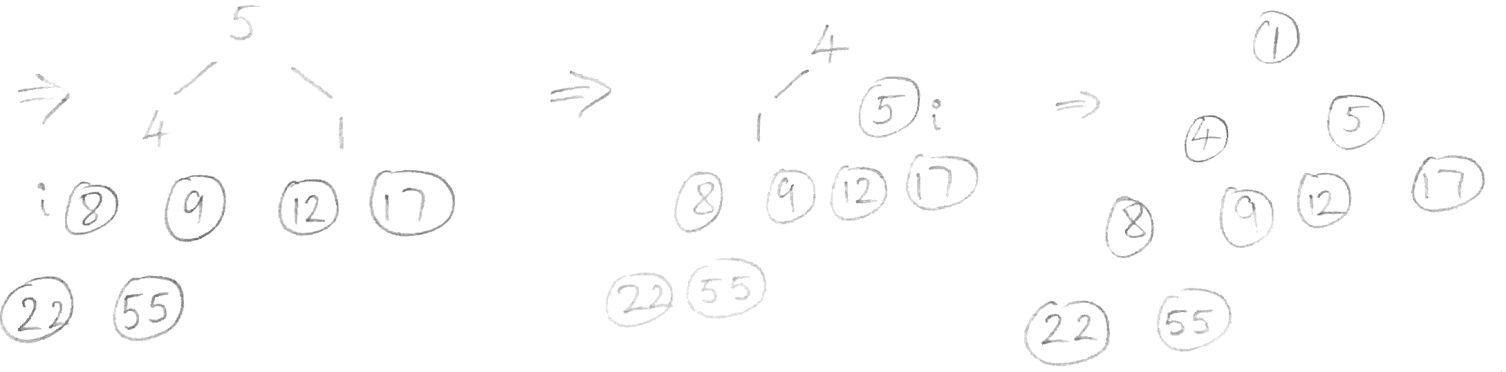
findBiggestElement (A, n)	Cost	# of executions
biggest = A[n]	C1	1
for i = $\frac{n}{2}$ + 1 to (n – 1)	C2	n – 1 – $\frac{n}{2}$ – 1 + 1 = $\frac{n}{2}$ – 1
if A[i] ≥ Biggest	C3	1
Biggest = A[i]	C4	1
return biggest	C5	1

$T(n) = C*1 + C2*(\frac{n}{2} - 1) + C3*1 + C4*1 + C5*1 = O(\frac{n}{2}) + C = O(n)$

④







✓

⇒ A

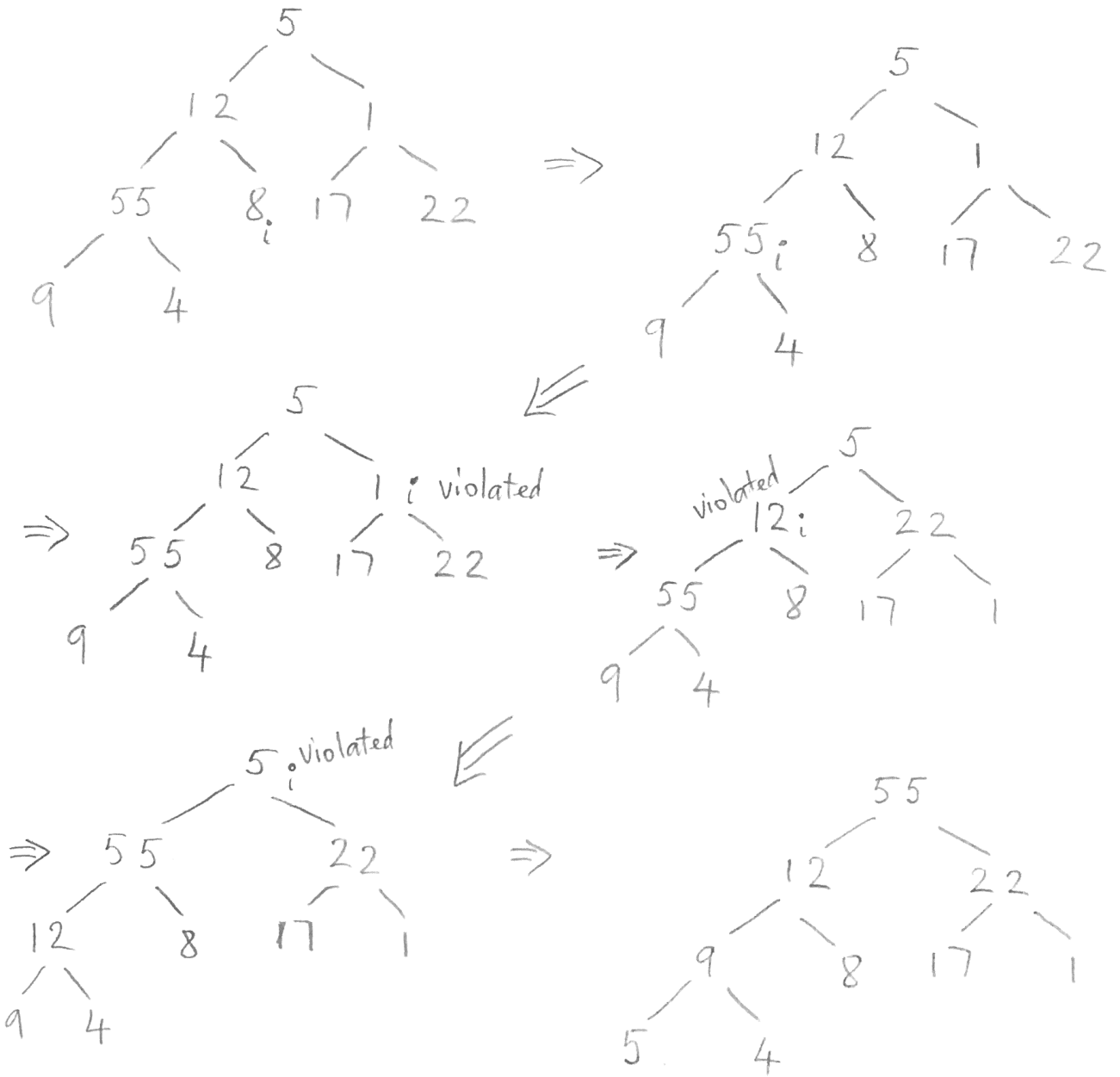
1	4	5	8	9	12	17	22	55

array A is sorted after HEAPSORT procedure finished.

Draft: HW6

④ $A = \{5, 12, 1, 55, 8, 17, 22, 9, 4\}$

BUILD_MAX_HEAP using figure 6.3 as a model (p.158)



Max heap:	2 children < parent
Min heap:	2 children > parent

5/

The operation HEAP-DELETE (A, i) deletes the item in node i from heap A. Give an implementation of HEAP-DELETE that runs in O(*log* n) time for an *n*-element max-heap.

(1) Pseudocode

		Cost	# of executions
	HEAP-DELETE (A, A.heap-size, i)		
(1)	A[i] = A[A.heap-size]	C1	1
	/ overwrites the value of A[i] by the value of the smallest leaf */*		
(2)	A.heap-size -= 1	C2	1
(3)	while (i > 1 and parent of i < A[i])	C3	1
(4)	swap (A[parent(i)]) with A[i]	C4	1
(5)	i = parent(i)	C5	1
(6)	MAX-HEAPIFY(A, i)	C6	1

(2) correctness justification

Done in the answer (1)

(2) provide an upper bound of your procedure and give an explanation

Explanation: as of A[i] has been removed and replaced by the smallest leaf, A[i] must be iteratively compared with its parent then swap if the parent is ≤ A[i] to ensure A[parent(i)] ≥ A[i] (line 3, 4, 5).

In case of A[i] is small, thus it recursively swaps with its child and traverses the longest path until it becomes a leaf.

$T(n) = C1*1 + C2*1 + C3*\sum_2^h 1 + C4*\sum_2^h 1 + C5*\sum_2^h 1 + C6*1$

$= C1 + C2 + (C3 + C4 + C5)*(h-2+1) + C6$

$= C*(h-1) + D$ (C and D are positive constants)

$= O(h) + D$ (provided $h = lg\ n$)

$= O(lg\ n)$ => proved

100/100