1) 10 Points

Compute an appropriately tight O (Big-Oh) bound on the running time of each code fragment, in terms of *n*. Assume integer arithmetic. Circle your answer for each fragment.

a)
```
for(i = 0; i < n; i++) {
     for(j = 0; j < i; j++) {
          for(k = 0; k < j; k++) {
               sum++;
          }
     }
}
```

$O(n^3)$

b)
```
for(i = 0; i < n; i++) {
     for(j = 0; j < n; j++) {
          if(i -- j) {
               for(k = 0; k < i; k++) {
                    sum++;
               }
          } else {
               for(k = 0; k < i * i; k++) {
                    sum++;
               }
          }
     }
}
```

$O(n^4)$

c)
```
for(i = n; i > 0; i = i / 2) {
     for(j = 0; j < n; j++) {
          sum++;
     }
}
```

$O(n \log n)$

d)
```
for(i = n; i > 0; i = i   (n / 2)) {
     for(j = 0; j < n; j++) {
          for(k = 0; k < n; k++) {
               sum++;
          }
     }
}
```

$O(n^2)$

2) 10 Points

Prove by induction that:

$n^2 \geq 2n$ for every integer $n \geq 2$

Base Case:

$n = 2 \qquad 2^2 = 4 \geq 4 = 2 * 2$

Inductive Hypothesis:

$n^2 \geq 2n$ for $2 \leq k \leq n$

Show: $(k + 1)^2 \geq 2(k + 1)$

$k^2 + 2k + 1 \geq 2k + 2$

$k^2 \geq 1$

$k^2 \geq 2k \geq 1$ for $k \geq 2$

3) 10 Points

Consider the *structure* of `tree a` and `tree b` (node keys are intentionally omitted).

```
tree a                                      tree b

       X                                           X
     /   \                                       /   \
   X       X                                   X       X
 /   \   /   \                               /   \
 X   X   X   X                               X   X
/ \     / \                                 / \
X X     X X                                 X X
```

For each question, answer "`tree a`", "`tree b`", "`neither`", or "`both`".

a) Which are perfect? neither

b) Which are complete? neither

c) Which are full? both

d) Which could be the structure of an array-based binary heap? neither

e) Which could be the structure of a leftist heap? both

f) Which could be the structure of a skew heap? both

g) Which could appear within a binomial queue? neither

h) Which could be the structure of a binary search tree? both

i) Which could be the structure of an AVL tree? tree a

j) Which could be the structure of a splay tree? both

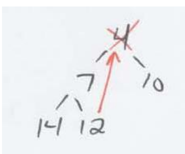
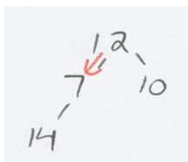
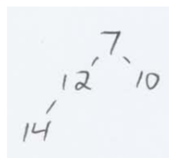
4) 10 Points

Consider the following binary min-heap:

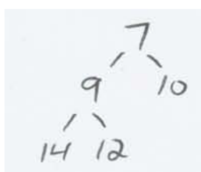
```
        4
      /   \
     7     10
    / \
  14   12
```

Perform the following operations in order, drawing the resulting heap after each operation and using it as the starting point for the next operation. You need only show the result of the operation, but showing your work will allow us to award partial credit. If the space here is insufficient, use the back of this sheet (clearly labeling your work). Circle the final state of the queue so we can distinguish it from intermediate work.

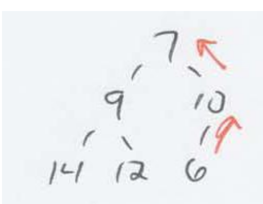
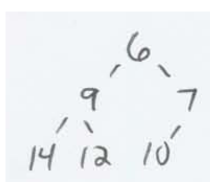
a) DeleteMin

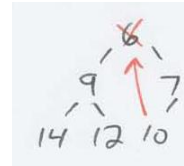
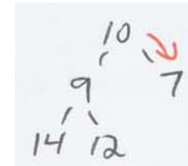
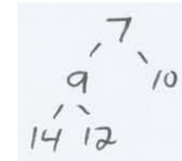
b) Insert 9

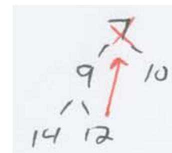
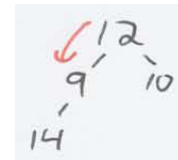
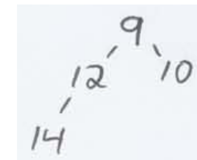
c) Insert 6

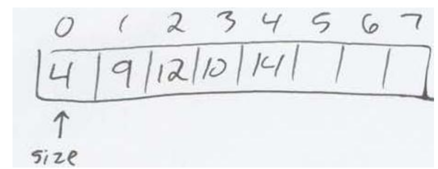
d) DeleteMin

e) DeleteMin

f) Draw an array-based representation of your heap from step *e*.



g) In your array-based representation, what is the index of:

the parent of the node at index *i*:

i / 2

the left child of the node at index *i*:

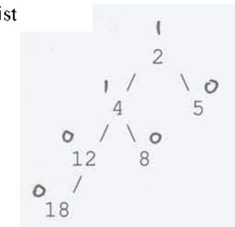
2i

the right child of the node at index *i*:

2i + 1

5) 10 Points

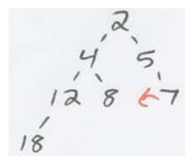
Consider the following leftist

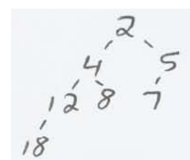


Perform the following operations in order, drawing the resulting heap after each operation and using it as the starting point for the next operation. You need only show the result of the operation, but showing your work will allow us to award partial credit. If the space here is insufficient, use the back of this sheet (clearly labeling your work). Circle the final state of the queue so we can distinguish it from intermediate work.

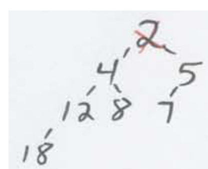
a) Annotate the above tree with the null-path length of each node.

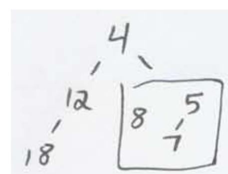
b) Insert 7

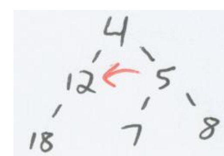


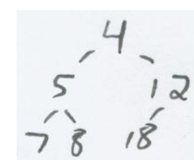


c) DeleteMin

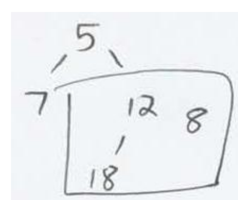


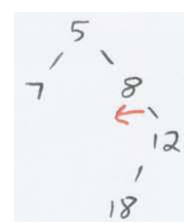


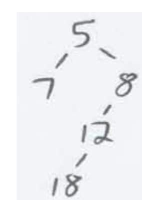




d) DeleteMin







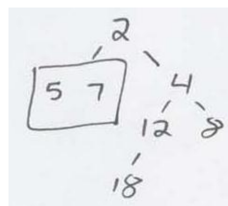
6) 10 Points

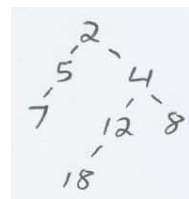
Consider the following skew heap:

```
          2
         / \
        4   5
       / \
     12   8
     /
   18
```

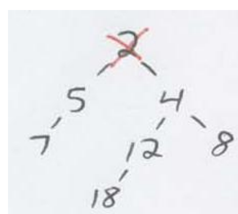
Perform the following operations in order, drawing the resulting heap after each operation and using it as the starting point for the next operation. You need only show the result of the operation, but showing your work will allow us to award partial credit. If the space here is insufficient, use the back of this sheet (clearly labeling your work). Circle the final state of the queue so we can distinguish it from intermediate work.

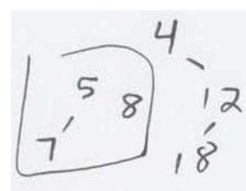
a) Insert 7

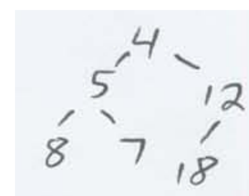




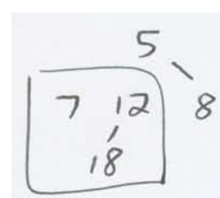
b) DeleteMin





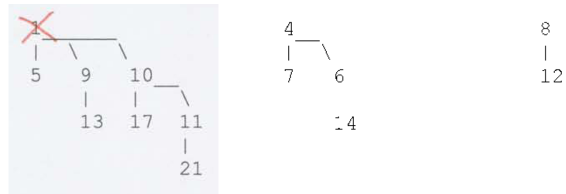


c) DeleteMin

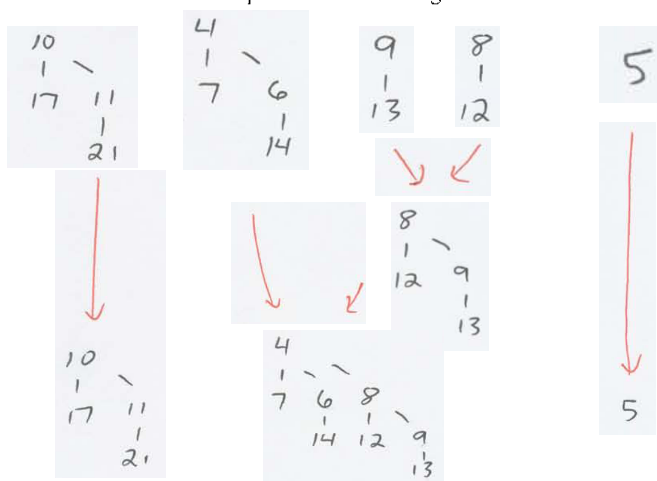


7) 10 Points

Consider the following binomial queue min-heap, which currently contains 3 trees:



a) Perform a deleteMin operation on this binomial queue. You need only show the result of the operation, but showing your work will allow us to award partial credit. If the space here is insufficient, use the back of this sheet (clearly labeling your work). Circle the final state of the queue so we can distinguish it from intermediate work.



b) If your answer for step *a* is correct, is it the only possible correct answer? Why?

No, made an arbitrary choice when choosing which two trees of size 4 to combine

c) If you merged four identical copies of your result from step *a*, the resulting binomial queue would contain how many trees of what size?

13 x 4 items = 52 items = 32 + 16 + 4

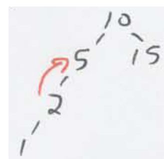
8) 10 Points

Consider the following AVL tree:

```
        10
       /  \
      5    15
     /
    2
```

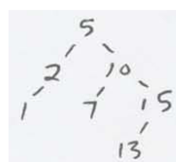
Perform the following operations in order, drawing the resulting tree after each operation and using it as the starting point for the next operation. You need only show the result of the operation, but showing your work will allow us to award partial credit. If the space here is insufficient, use the back of this sheet (clearly labeling your work). Circle the final state of the queue so we can distinguish it from intermediate work.

a) Insert 1

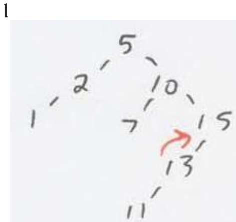


b) Insert 7

c) Insert 13



d) Insert 11



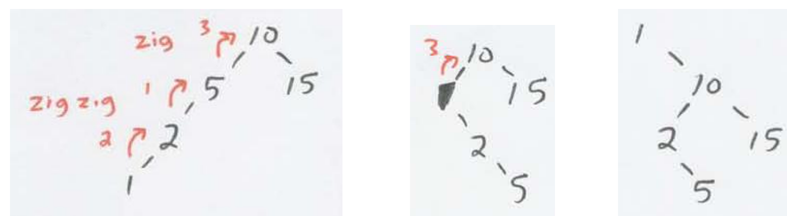
9) 10 Points

Consider the following splay tree:

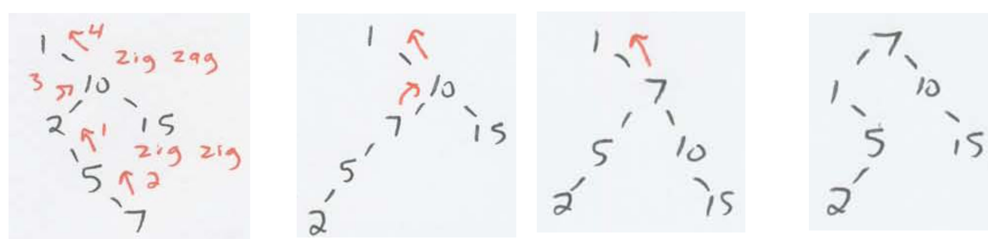
```
      10
     /  \
    5    15
   /
  2
```

Perform the following operations in order, drawing the resulting tree after each operation and using it as the starting point for the next operation. You need only show the result of the operation, but showing your work will allow us to award partial credit. If the space here is insufficient, use the back of this sheet (clearly labeling your work). Circle the final state of the queue so we can distinguish it from intermediate work.

a) Insert 1



b) Insert 7



c) Delete 5