Key

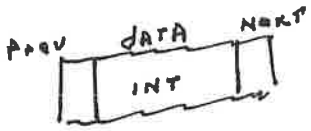
Name Key **Andrew ID** Key

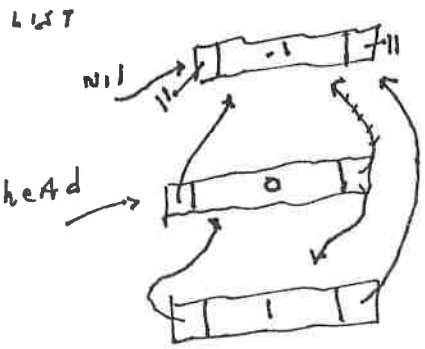
Total points = ~~131~~ 133. Score will be a percentage of ~~131~~ 133.

Tracing Lists and Trees (~~28~~ 30 points)

1. You will be asked to show the exact output of the following program. (~~18~~ 20 Points)

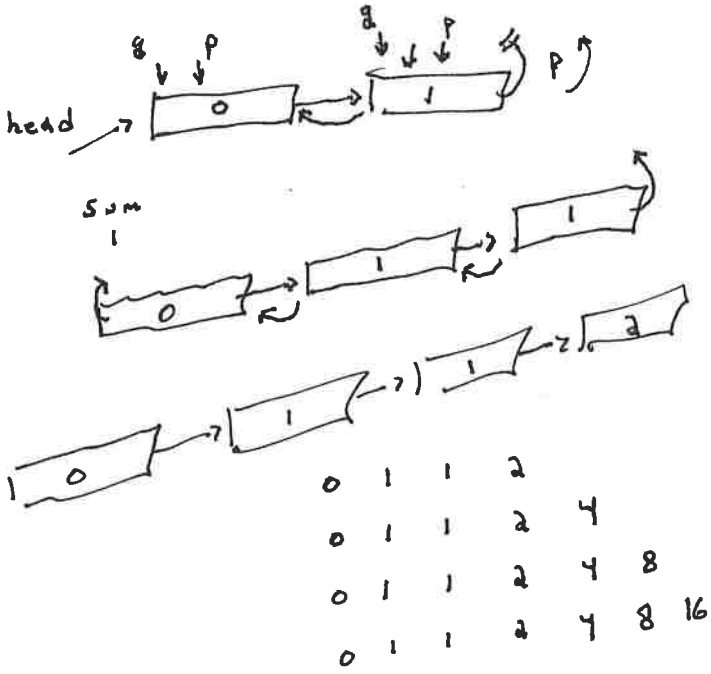
```
package org.example;

class Node {
   private int data;
   private Node next;
   private Node prev;

   public int getData() {
      return data;
   }
   public void setData(int data) {
      this.data = data;
   }

   public Node getNext() {
      return next;
   }
   public void setNext(Node next) {
      this.next = next;
   }

   public Node getPrev() { return prev; }
   public void setPrev(Node prev) {
      this.prev = prev;
   }

   public Node(Node prev, int data, Node next) {
      this.data = data;
      this.next = next;
      this.prev = prev;
   }
}
class List {
   Node nil;
   Node head;
   public List() {

      nil = new Node(null,-1,null);
```

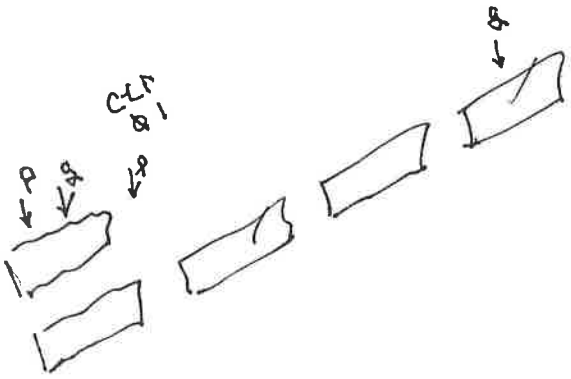




Key 2

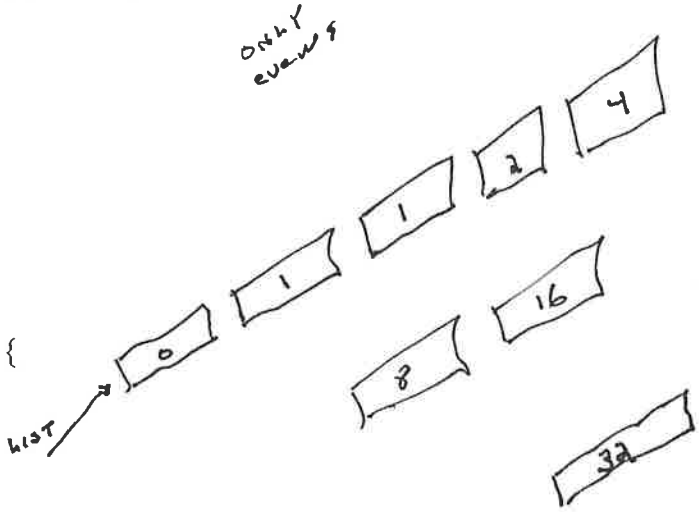
```
    head = new Node(nil,0,nil);
    head.setNext(new Node(head,1,nil));

  }
  // (1.a) Big Theta:
  public void add() {
     int sum = 0;
     Node p = head;
     while (p != nil) {
        sum = sum + p.getData();
        p = p.getNext();
     }
     p = head;
     Node q = p;
     while(p != nil) {
        q = p;
        p = p.getNext();
     }
     q.setNext(new Node(q,sum,nil));
  }

  public boolean isEmpty () {
     return head == nil;
  }

  // (1.b) Big Theta:
  public String toString() {
     Node v = head;
     String s = "";
     while(v != nil) {
        s = s + v.getData() + " ";
        v = v.getNext();
     }
     return s;
  }
  // (1.c) Big Theta:
  public String traverse() {
     String result = "";
     Node p = head;
     Node q = p;
     int ctr = 0;
     while (p != nil) {
        q = p;
        p = p.getNext();
        ctr = ctr + 1;
     }
```





Key 3

```
    while(q != nil) {
      if(ctr % 2 == 0) result = result + q.getData() + " ";
      q = q.getPrev();
      ctr = ctr - 1;
    }
    return result;
  }
}

public class DSMidterm {

  public static void main(String[] args) {
    List list = new List();
    list.add();
    list.add();
    list.add();
    list.add();
    list.add();
    // 1.d Show output of println
    System.out.println(list);
    // 1.e Show output of println
    System.out.println(list.traverse());
    list.add();
    // 1.f Show output of println
    System.out.println(list);
    // 1.g Show output of println
    System.out.println(list.traverse());


  }
}
```



0 1 1 2 4 8 16

8 2 1

0 1 1 2 4 8 16 32

32 8 2 1

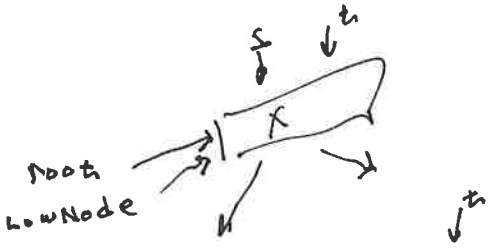
1.(a) Give the Big Theta of the code marked (1.a) (add). $\Theta(1)$ (2 pts)
1.(b) Give the Big Theta of the code marked (1.b) (toString). $\Theta(n)$ (2 pts)
1.(c) Give the Big Theta of the code marked (1.c) (traverse). $\Theta(n)$ (1 pts)
1.(d) Show the output of the code marked (1.d). 0 1 1 2 4 8 16 (1 pts)
1.(e) Show the output of the code marked (1.e). 8 2 1 (4 Points)
1.(f) Show the output of the code marked (1.f). 0 1 1 2 4 8 16 32 (4 Points)
1.(g) Show the output of the code marked (1.g). 32 8 2 1 (4 Points)

1.(h) Is it correct to say that the method add() of the list class runs in $O(2^n)$?
Circle True or False (1 pt.)

1.(i) Is it correct to say that the method traverse() of the list class runs in $\Omega(n^2)$?
Circle True or False (1 pt.)

2. Trace the execution of the following program. Five questions appear below. (10 points):


```
class Node {
   public int data;
   public Node lc;
   public Node rc;
   public Node p;
   public Node(Node lc, int x, Node rc, Node p) {
      this.lc = lc;
      this.data = x;
      this.rc = rc;
      this.p = p;
   }
}
public class SimpleTree {

   public Node root;
   public Node lowNode;

   public SimpleTree() {
      root = null;
      lowNode = null;
   }

   public void add(int x){

      if (root == null) {
         root = new Node(null,x,null,null);
         lowNode = root;
      }
      else {
         Node t = root;
         Node q = t;
         while(t != null) {
            if(x < t.data ) {
               q = t;
               t = t.lc;
            }
            else {
               q = t;
               t = t.rc;
            }
         }
         if(x < q.data) {
```

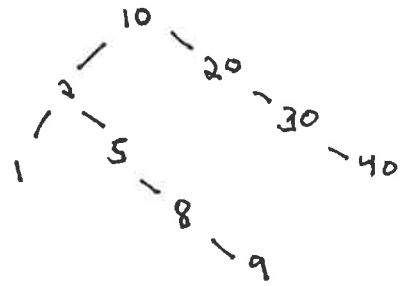


Key's

```
            q.lc = new Node(null, x, null, q);
            lowNode = q.lc;
         }
         else {
            q.rc = new Node(null, x, null, q);
            lowNode = q.rc;
         }
      }
   }
   public void traversal(Node r) {
      if(r == null) return;
      if(r.lc != null)traversal(r.lc);
      if(r.rc != null)traversal(r.rc);
      System.out.println(r.data);
   }
   public void traversal() {
      traversal(root);
   }

   public void traversal2() {
      Node q = lowNode;
      while( q != null) {
         System.out.println(q.data);
         q = q.p;
      }
   }
   public static void main(String[] args) {
      SimpleTree st = new SimpleTree();
      st.add(10);
      st.add(20);
      st.add(30);
      st.add(40);
      st.add(2);
      st.add(5);
      st.add(8);
      st.add(9);
      System.out.println("(2.a)");
      st.traversal();
      System.out.println("(2.b");
      st.traversal2();
      st.add(1);
      System.out.println("(2.c)");
      st.traversal2();

   }
}
```

Postorder

recent leaf to root



2.A 9 8 5 2 40 30 20 10

2.B 9 8 5 2 10

2.C 1 2 10

2.(a) What will the program display at the traversal marked Question 2.a? (2 Points)

9 8 5 2 40 20 20 10

2.(b) What will the program display at the traversal marked Question 2.b? (2 Points)

9 8 5 2 10

2.(c) What will the program display at the traversal marked Question 2.c? (2 Points)

1 2 10

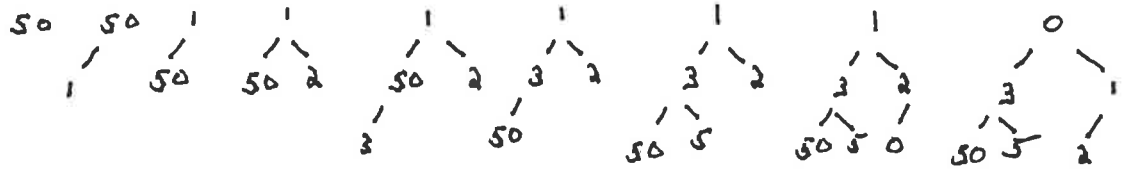
2.(d) What is the worst case Big Theta value for traversal2(). Our only assumption is that the tree holds n nodes in total. $\Theta(N)$ (2 points)

2.(e) Suppose we built a CLR complete tree and it was perfectly balanced (unlike the tree created above). Suppose too that the tree held n nodes in total. Provide a Big Theta value for traversal(). $\Theta(N)$ (2 points) VISITS EACH NODE

Heaps (12 points)

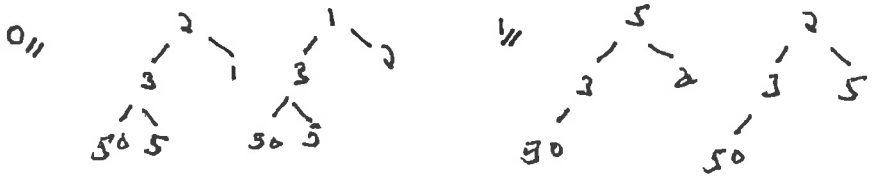
3) Insert the following 6 numbers into a min heap. Draw a new tree for each heap insertion. (4 Points)

50, 1, 2, 3, 5, 0



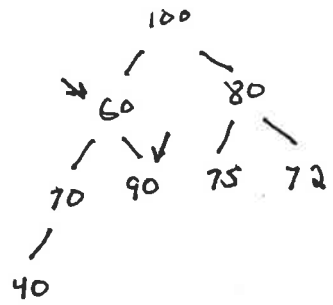
4) What is the height of the tree that you drew in question 3? (A single node in a tree gives a height of 0.)
(2 Points) 2

5) Perform exactly two deleteMin() operations on the heap that you drew in question 3. Draw the resulting trees. Make it clear to the reader what is going on. (3 Points)



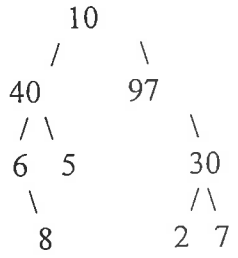
6) Consider the following max heap implemented in an array. It is not quite correct. To make it a proper max heap exactly one swap must occur. What two numbers (child and parent) need to be swapped in order to make this a max heap? (3 points). PLACE CHECK MARKS NEXT TO THE TWO NUMBERS THAT NEED TO BE SWAPPED.

100
✓ 60
80
70
✓90
75
72
40



Trees (16 points)

7. Parts (a), (b), (c) refer to the following binary tree:



(a) List the data that would be accessed by a pre-order traversal on the given tree by writing out the values in the nodes as they would be accessed, separated by commas. (3 points)

10 40 6 8 5 97 30 2 7

(b) List the data that would be accessed by an in-order traversal on the given tree by writing out the values in the nodes as they would be accessed, separated by commas. (2 points)

6 8 40 5 10 97 2 30 7

(c) List the data that would be accessed by a level-order traversal on the given tree by writing out the values in the nodes as they would be accessed, separated by commas. (2 points)

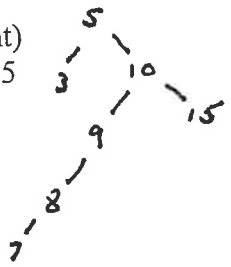
10 40 97 6 5 30 8 2 7

(d) In general, if a ternary tree (at most three children per node) is perfectly balanced (unlike the tree pictured above) and complete with height h, how

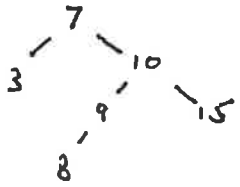
many leaves will the tree have? (2 points) 3^h Note, this tree has a perfectly flat bottom. We need the total number of leaves in terms of h. This is an exact answer, not Big O.

(e) In general, if a ternary tree (at most three children per node) is perfectly balanced (unlike the tree pictured above) and complete with exactly k leaves. What is the height (in terms of k) of this tree? (2 points) $\log_3 K$ Note, this tree has a perfectly flat bottom. This is an exact answer, not Big O.

8. (a) Insert he following numbers into a Binary Search Tree. Draw the tree after all insertions are complete. (1 Point)
5, 3, 10, 9, 8, 7, 15



(b) Delete 5 from the final tree that you drew in 8 (a). We are following the "go right once, left hard" rule. Draw this final tree. (1 Point)



(c) Delete 7 from the final tree in question 8 (b). Again, we are following the "go right once, left hard" rule. Draw this final tree. (3 Points)

Keya

Project Questions (20 points)

(9) Recall the Merkle-Hellman cryptosystem that we worked with in Project 1.

Project 1 was based on the subset sum problem which is known to be NP-Complete. The problem itself can be described as follows: given a set of numbers X and a number **k**, is there a subset of X, which sums to k?

(a) Suppose X = {100, 9, 2, 105, 3, 7, 101} and **k** = 14. Is there a subset of X which sums to k?

Yes (Yes) No (1 point)

(b) The type of problem you were asked to solve in question 9 (a) is (Circle one answer): (1 Point)

1. a problem that is impossible to solve.
2. a problem that has been proven to take exponential time to solve.
3. a problem that has been proven to take factorial time to solve.
4. a decision problem.
5. an optimization problem.

(c) Suppose Alice sends a message (K) to Bob. K is computed using Bob's Merkle-Hellman public key combined with the message M. The central idea behind Merkle-Hellman is that a potential eaves dropper could read the message M if the eaves dropper could (circle the one best option) (1 Points)

1. Find a subset of Bob's public key that sums to K.
2. Find K so that M is prime.
3. Modify Bob's public key.
4. Modify the super increasing sequence.
5. Find a subset of a super increasing sequence that sums to K.

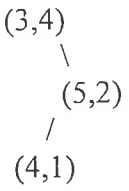
$3 \cdot 1 \equiv 3$
$3 \cdot 2 \equiv 6$
$3 \cdot 3 \equiv 2$
$3 \cdot 4 \equiv 12 \equiv 5$
$3 \cdot 5 \equiv 15 \equiv 1$

(d) Recall that a modular inverse of an integer b mod m is the integer b^{-1} such that $(b * b^{-1}) \bmod m = 1$.

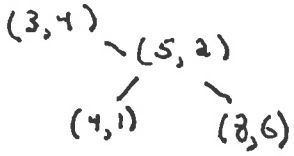
What is the modular inverse of 3 mod 7? 5 (2 Points)

(e) The following points, in a standard (x,y) coordinate plane, have been added to a 2-d tree. (3,4), (5,2), (4,1). The 2-d tree appears as follows:

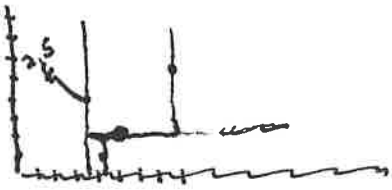
Key 10



Add the point (8,6) to this 2-d tree. Redraw the tree with this new point added. The first point, (3,4), breaks the plane vertically. (2 points)



(f) Consider the 2-d tree that you created, with the addition of (8,6), in (e). Suppose that we performed a nearest neighbor search for the point (2,5). Which points in the tree need to be examined? (3,4), (5,2) (8,6)
(2 Points)



(g) In Project 3 we wrote a Red Black binary search tree. Suppose we are doing a lookup for a course name in the Red Black Tree. Let T(n) be the number of operations required to do the lookup. In the worst case, which of the following are true about T(n)? Circle all of those that are true. (You may or may not have more than one answer.) (4 Points)

1. $T(n) \in O(3^n)$
2. $T(n) \in \Omega(n^2)$
3. $T(n) \in O(n)$
4. $T(n) \in \Theta(\text{Log}n)$
5. $T(n) \in O(2^n)$
6. $T(n) \in \Theta(n)$
7. $T(n) \in O(n!)$
8. $T(n) \in \Omega(n)$
9. $T(n) \in O(\text{Log}(n))$

(h)The following is a data file for Project 3. Note the course Philo2 that is taken by Bill. The purpose of the Red-black tree was to maintain an integer with each course name. What integer will be assigned to Bill's Philo2 in the Red-Black tree of Project 3? (4 Points) 2

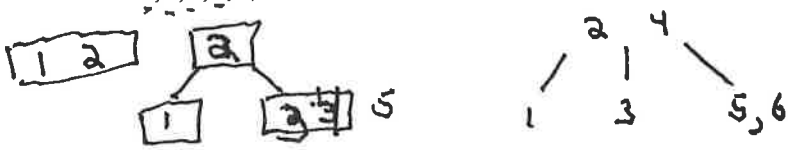
Amy Calc1 Span1 Philo2
Bill Calc1 Philo2 Hist4

key 11

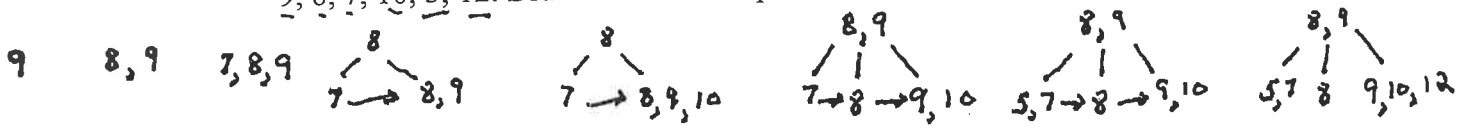
(i) Consider the graph generated by the Project 3 dataset in question 9 (h). We are referring to the previous question. How many edges will the node labelled "Hist4" have? (3 Points) 2

B Trees (21 points)

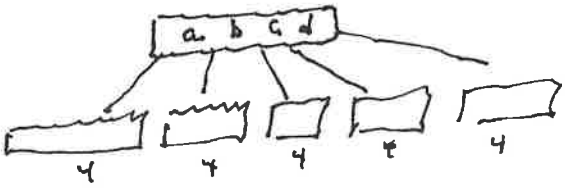
10. (a) Insert the following numbers into a B-Tree with a minimum of 1.
1,2,3,4,5,6 Draw the final tree. (7 Points)



(b) Insert the following numbers into a B+ Tree with a minimum of 1.
9, 8, 7, 10, 5, 12. Draw each tree for partial credit. Draw the final tree. (7 Points)



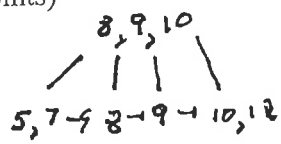
(c) Consider a B-Tree with a minimum of 2. What is the exact maximum number of keys such a tree could hold if the tree were of height 1? 24 (7 Points)

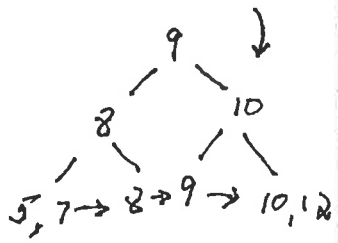


4 keys

20 keys

24 keys





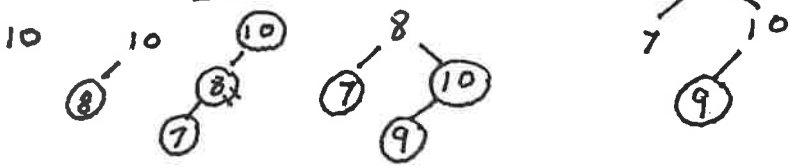
Key, 2

Red Black Trees (8 points)

11. Red Black Trees

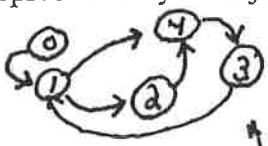
(a) Insert the following numbers, one by one, into a Red-Black Tree. Show the tree after each insertion. Draw RED nodes with a circle or a label 'R'. (8 points)

10, 8, 7, 9



Graph Representations (8 points)

Cnsider the directed graph G_1 . The graph is represented by an adjacency matrix m. If there is an edge from i to j then m[i,j] = true.



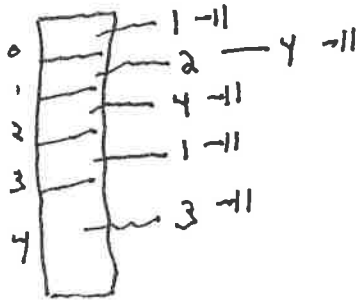
Matrix m

vertex	0	1	2	3	4	5
0		true				
1			true		true	
2					true	
3		true				
4				true		

G_1

12. (a) Draw the graph G_1 with circles and edges. (2 Points)

12. (b) Suppose that we decide to represent the graph G_1 with an adjacency list rather than an adjacency matrix. Draw a picture of what that representation would look like. (6 Points)



Key 13

Big O (18 Points)

13. I have just arrived by plane at Pittsburgh International Airport and I have forgotten where I have parked my car. There are n cars in the parking lot. I decide to look for my car with a simple serial search, examining, one by one, each and every car in the lot. This algorithm could be analyzed by considering its best case, average case, and worst case. Circle each of the following answers that are mathematically correct. (3 Points)

(a.) In the worst case, the algorithm is Big Omega(1).
(b.) In the best case, the algorithm is Big O(Log n).
(c.) In the best case, the algorithm is Big O(n).
d. In the worst case, the algorithm is Big Theta(n^2).
e. In the best case, the algorithm is Big Theta(n).

14. I have just arrived by plane at Pittsburgh International Airport and I have forgotten where I have parked my car. There are n cars in the parking lot. I have a new IoT gadget in my pocket. I click on the gadget and my car begins to beep loudly. I hear my car and walk directly to it. This algorithm (circle all that are correct). (3 Points)

(a.) runs in Big O(1).
(b.) runs in big O(n).
(c.) runs in Big Theta(1).
(d.) runs in Big Omega(1).
e. runs in Big Omega(n).

15. I have just arrived by plane at Pittsburgh International Airport and I have forgotten where I have parked my car. There are n cars in the parking lot. Suppose that we have great parking lot attendants and the cars have been sorted by time of arrival. The car that has arrived most recently is at the very end of the line. The car that has been in the lot the longest is at the front of the line. I walk to the middle of the line and I am able to determine the time of arrival for the middle car (another IoT gadget). I then either choose the back half of the line or the front half of the line and walk there - again, going to the middle car. Circle all of the mathematically correct answers: (3 Points)

(a.) In the worst case, this algorithm is Big Omega(1).
(b.) In the best case, this algorithm is Big Theta(1).
c. In the worst case, this algorithm is Big Omega(n).
(d.) In the best case, this algorithm is Big O(n).
(e.) In the worst case, this algorithm is Big Theta(Log n).

16. I work as a parking lot attendant at the Pittsburgh International Airport and my job is to keep all of the cars in the lot in order by time of arrival. It's a strange job but it pays well. When a new car arrives, I simply park it at the end of the line. The real work comes in when someone picks up their car and leaves. I have to move each car behind the new empty slot one slot forward. There are n cars in the lot. This shifting of cars has (circle all that are correct). (4 Points)

Key 14

(a.) a worse case of big O(2^n)
b. a worse case of big Theta(n^2).
(c.) a worse case of big Theta(n).
(d.) a best case of big Theta(1).
(e.) a worse case of big O(n^2).
(f.) an average case of big Theta(n).

17. I work as a car thief at the Pittsburgh International Airport and my job is to steal every single car in the lot of n cars. It takes me about five minutes per car to break the lock and another 20 minutes to drive the car away, and get back for the next car. In other words, it takes a constant amount of time to break the lock and another constant amount of time to get back to the next car. Circle all of the following that are mathematically true: (5 Points)

(a.) This job has no cases to consider and is Big O(n^2).
(b.) This job has no cases to consider and is Big Theta(n).
(c.) This job has no cases to consider and is Big Omega(1).
d. This job has no cases to consider and is Big Omega(n^2).
e. This job has no cases to consider and is Big Theta(1).