```
public class Test
{  int data;
   public Test(int x )    { data=x; }
   public void assign(Test b, Test c, int d)
   {   data= 2;
       b= new Test(2);
       c.data= 2;
       d=2;              }
   public static void main(String [] args)
   {   Test a, b, c;    int d;
       a= new Test(1); b= new Test(1);
       c= new Test(1); d=1;
       a.assign(b, c, d);
       System.out.println(a.data + " " + b.data + " "
                               + c.data + " " + d);     }        }
```

What does this print?

```
public class Test
{
   int data;

   public Test(int x )
   {
        data=x;
   }

   public void assign(Test b, Test c, int d)
   {
        data= 2;
        b= new Test(2);
        c.data= 2;
        d=2;
   }
```

```
public static void main(String [] args)
{
    Test a, b, c;
     int d;

      a= new Test(1);
      b= new Test(1);
      c= new Test(1);
      d=1;

      a.assign(b, c, d);

      System.out.println(a.data + " " +
                         b.data + " " +
                         c.data + " " +
                         d);
 }

}
```

On slides: I may use awkward spacing in order to allow visibility/increased font size. Please format your code nicely! Also, include lots of comments!

Announcements

The official course outline has been posted.

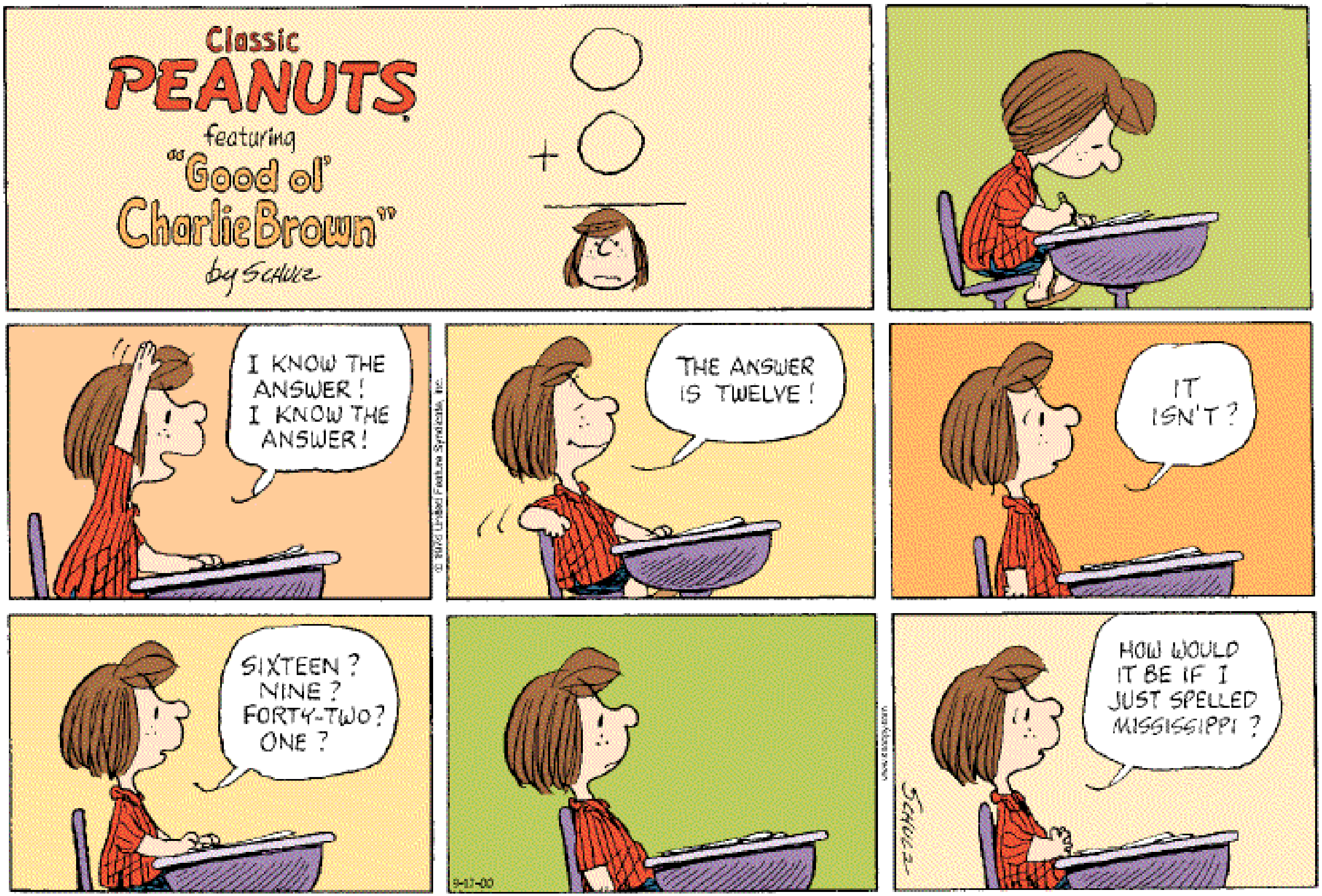
Any questions about the course?

Make sure you sign the attendance sheet, BUT do not sign or have anyone else sign for you if you do not plan to attend the whole class.

Assignment #1A and 1B and Tutorial #1 are posted. Tutorials start next week.

Bring your schedule to class on Wednesday (to help me in choosing office hours).

You will learn a lot more if you try the problems and get them wrong than if you do not try. Also, it helps me know where the class is in terms of understanding.


Classic
PEANUTS
featuring
"Good ol' Charlie Brown"
by SCHULZ
+
I KNOW THE ANSWER! I KNOW THE ANSWER!
THE ANSWER IS TWELVE!
IT ISN'T?
SIXTEEN? NINE? FORTY-TWO? ONE?
HOW WOULD IT BE IF I JUST SPELLED MISSISSIPPI?
9-17-00
SCHULZ

Slide by: Kevin Wayne
Princeton University

Why study algorithms?

Their impact is broad and far-reaching.

Internet. Web search, packet routing, distributed file sharing, ...

Biology. Human genome project, protein folding, ...

Computers. Circuit layout, file system, compilers, ...

Computer graphics. Movies, video games, virtual reality, ...

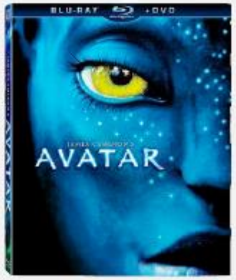
Security. Cell phones, e-commerce, voting machines, ...

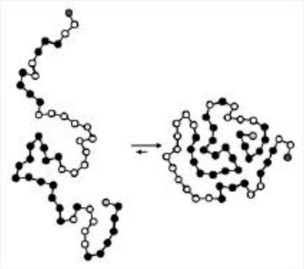
Multimedia. MP3, JPG, DivX, HDTV, face recognition, ...

Social networks. Recommendations, news feeds, advertisements, ...

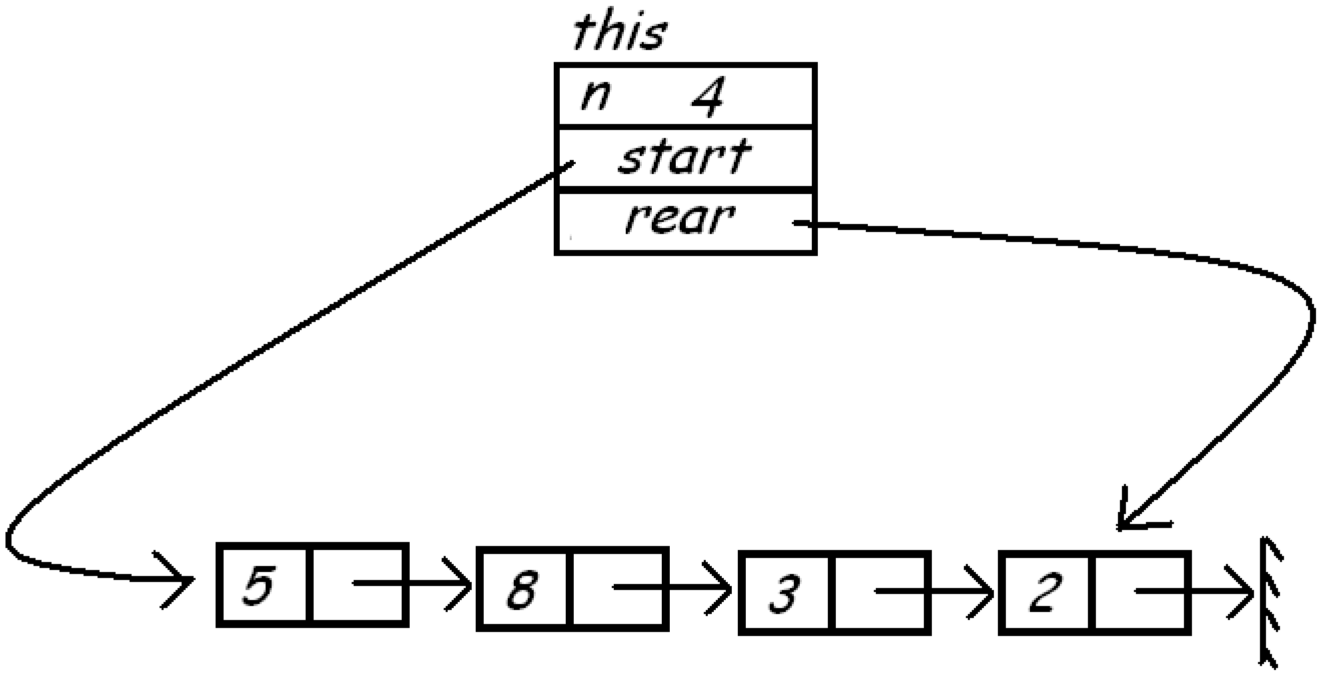
Physics. N-body simulation, particle collision simulation, ...

...





Review of linked lists



How can we split a list into two lists for our Mergesort?

One solution: evenOddSplit()

Side note: this is not the right way to do it on the assignment.

Learning objectives: Understanding of how to program linked lists in Java. To learn how to draw pictures of what a program is doing as it is executing (this will help you to more easily write correct code and to debug code which is not correct).

Testing evenOddSplit:

Walk through some test examples by hand to make sure you know how to step through code and draw pictures of what is happening as it is executing.

Programs most often have mistakes with extremal cases. What happens with lists with 0, 1, and 2 items?

The code is doing different things with even numbered and odd numbered cells of the list. Does it work properly for lists of even length? Does it work properly for lists of odd length?

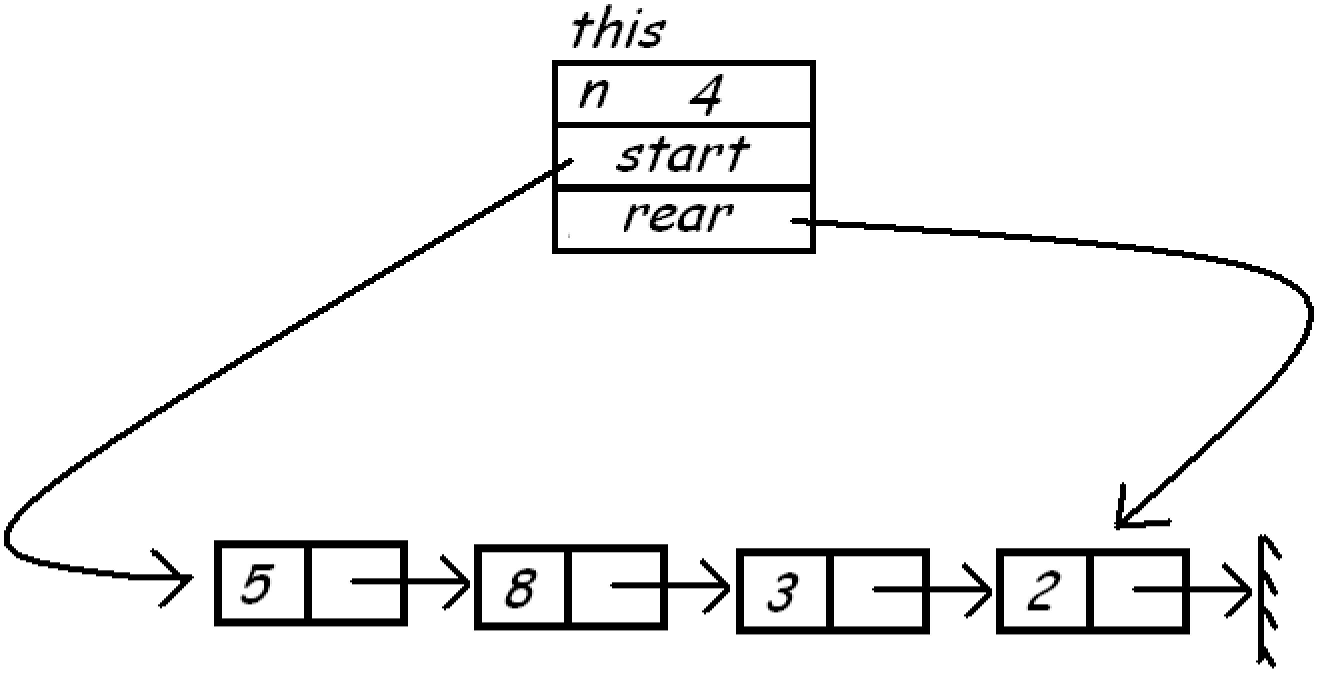
Fix any bugs you find.

Using the data structures for the lab:

```
class ListNode{

   public int data;
   public ListNode next;

   public ListNode(int x, ListNode ptr)
   {
      data= x;
      next= ptr;
   }
}
```

```
public class LinkedList
{
   int n;
   ListNode start;
   ListNode rear;

   public LinkedList()
   {
      n= 0;
      start= null;
      rear= null;
   }
```

```
public class SplitList
{
   LinkedList list1;
   LinkedList list2;

   public SplitList()
   {
      list1= new LinkedList();
      list2= new LinkedList();
   }
}
```



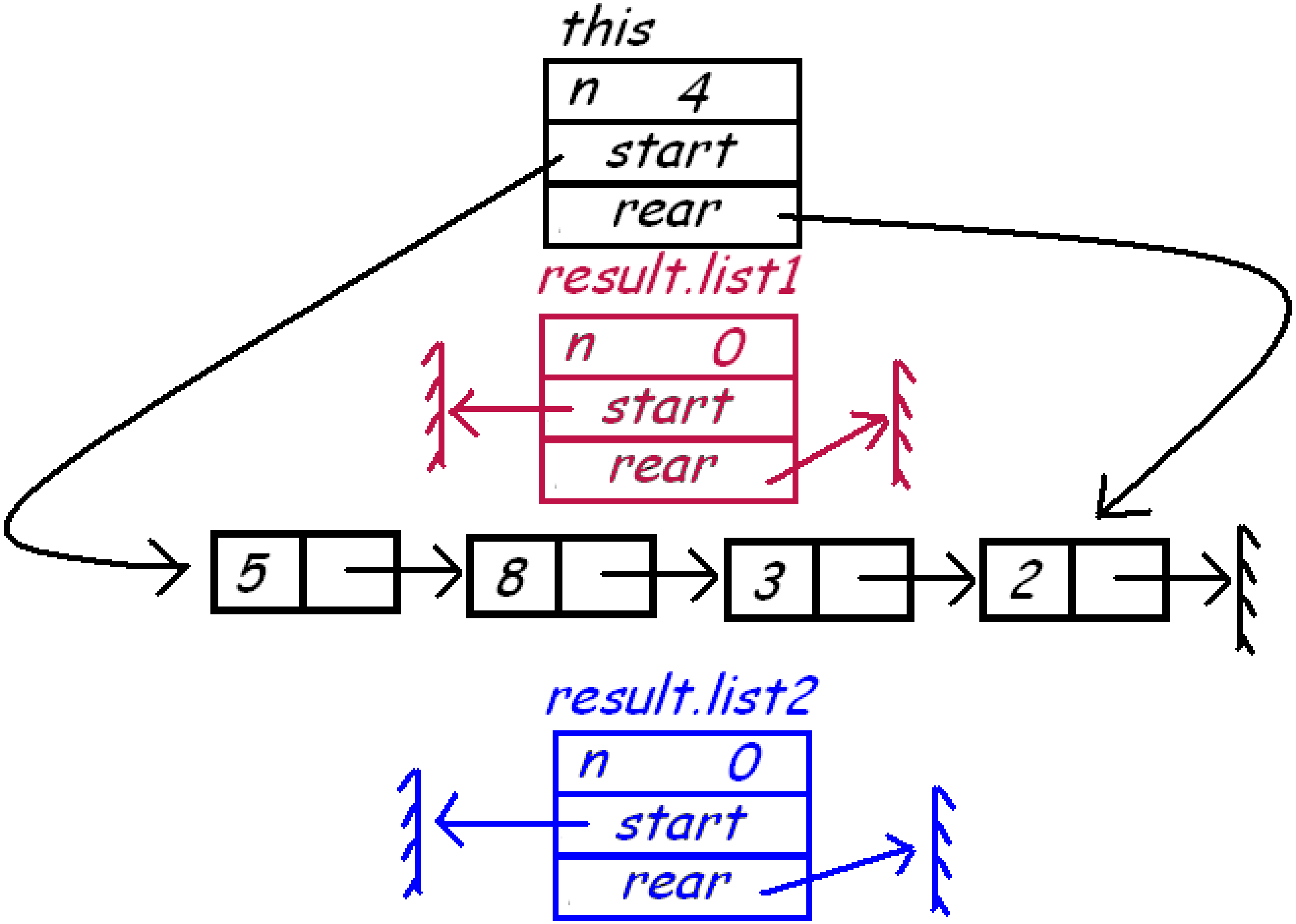
Step through the code in evenOddSplit() with this example.

```
public SplitList evenOddSplit()
{    SplitList result;

     result= new SplitList();
     if (start== null) return(result);
     result.list1.start= start;
     result.list1.rear = start;
     result.list1.n++;

     if (start.next == null) return(result);
     result.list2.start= start.next;
     result.list2.rear = start.next;
     result.list2.n++;
```

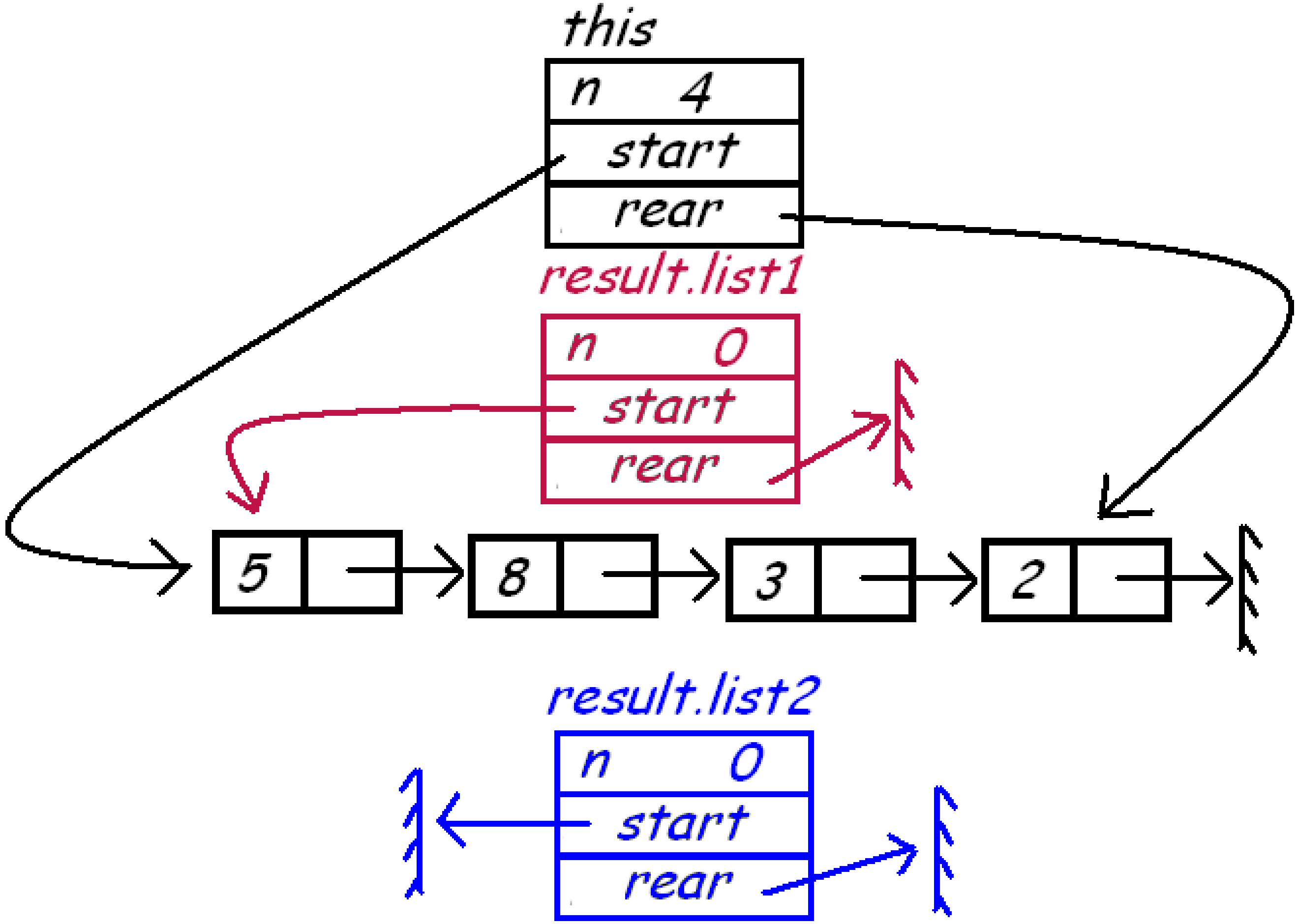
Buggy
code

```
while (result.list2.rear.next!= null )
{
  result.list1.rear.next= result.list2.rear.next;
  result.list1.rear= result.list2.rear.next;
  result.list1.n++;
  if (result.list1.rear.next != null)
  {
     result.list2.rear.next= result.list1.rear.next;
     result.list2.rear= result.list1.rear.next;
     result.list2.n++;
   }
}
```

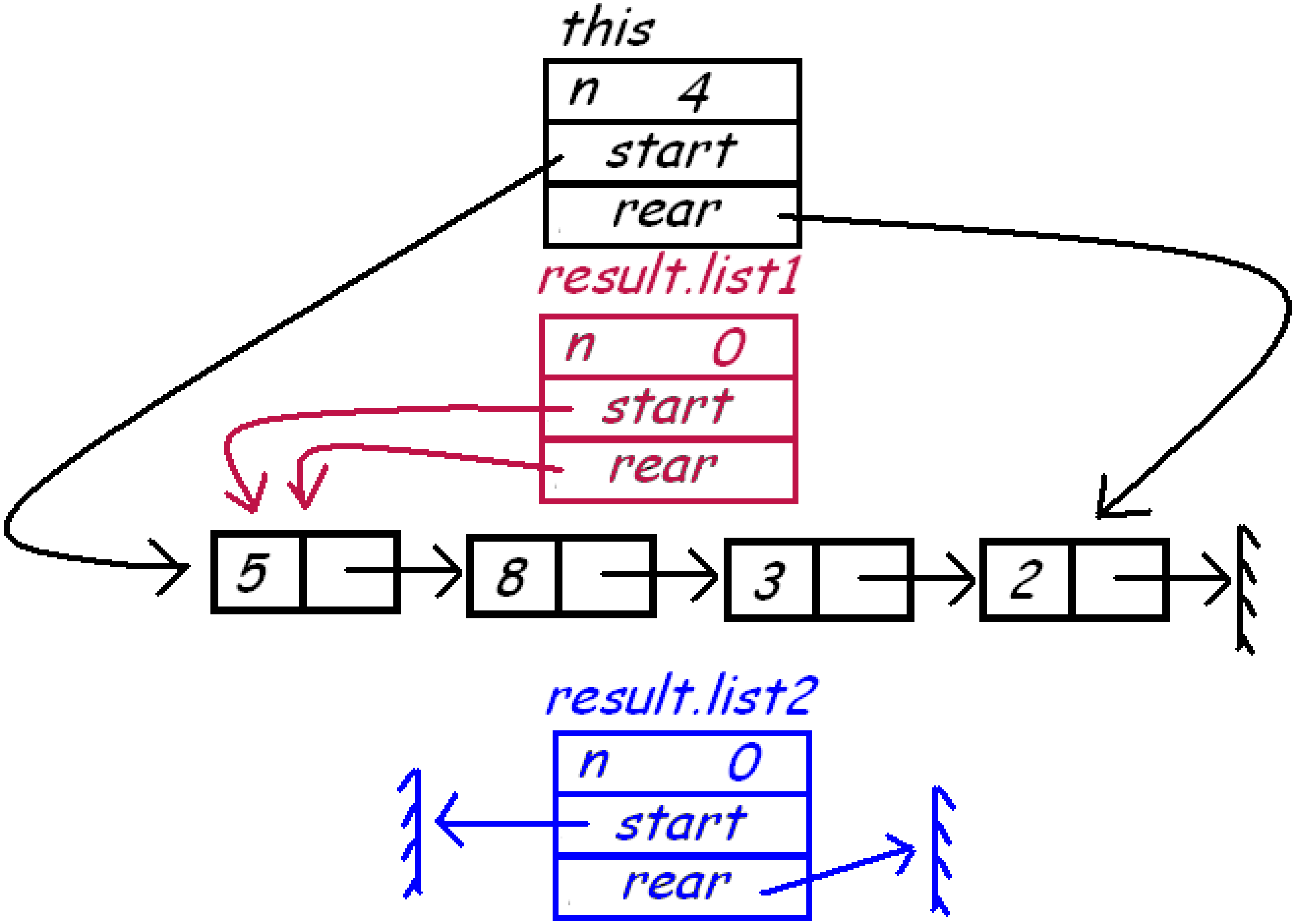
```
result.list1.rear.next= null;
result.list2.rear.next= null;
return(result);
```



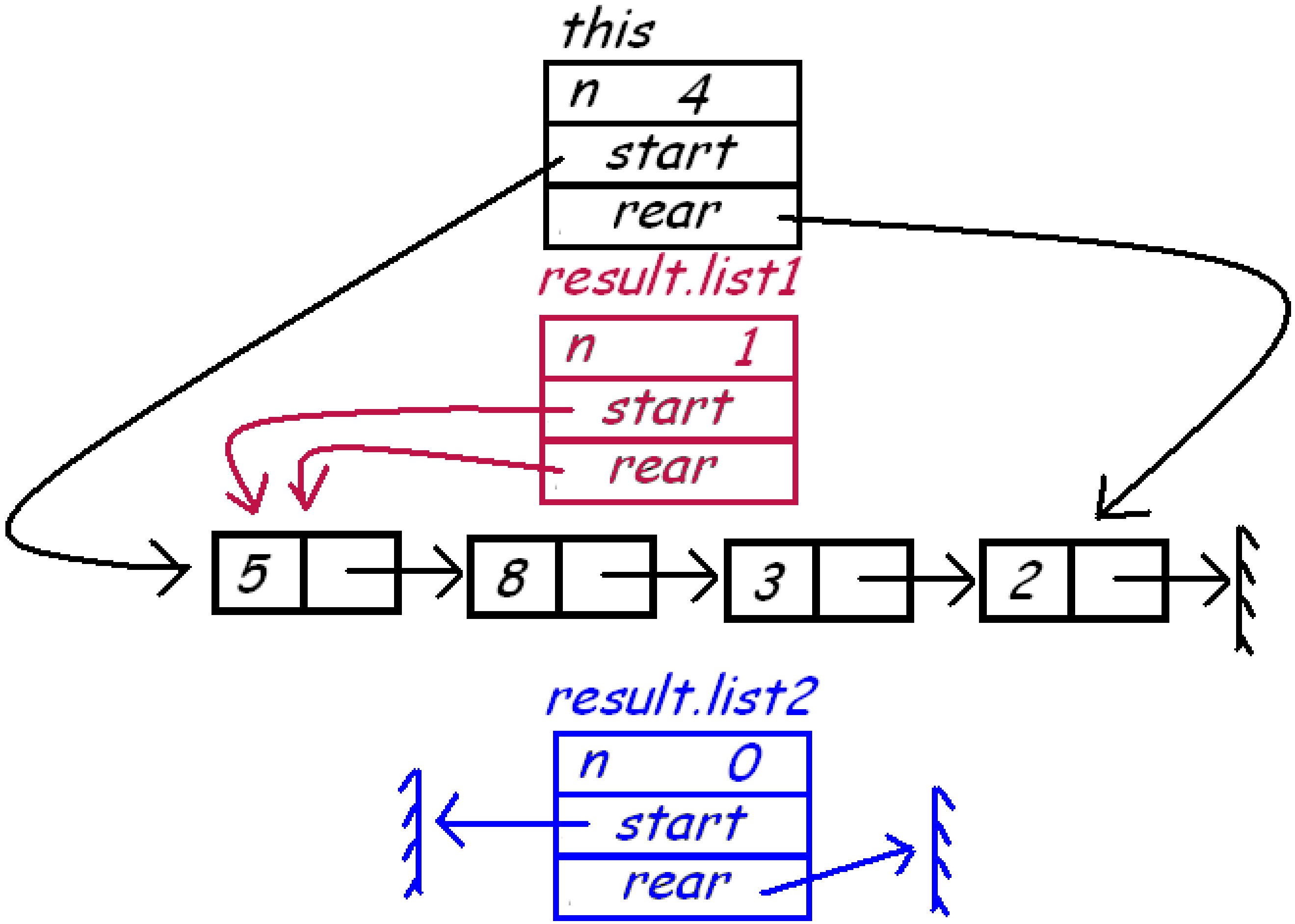
```
result= new SplitList();
```



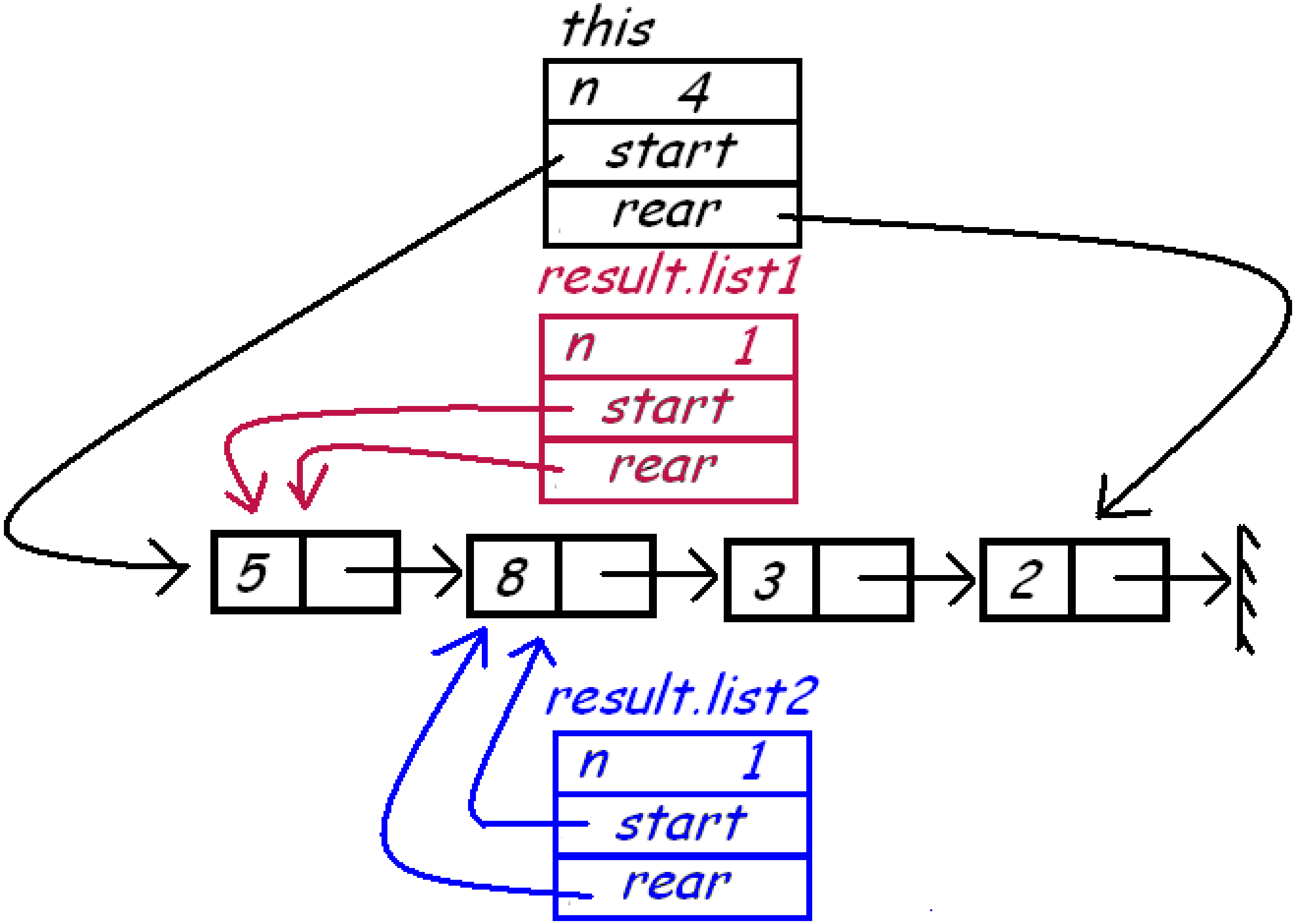
result.list1.start= start;



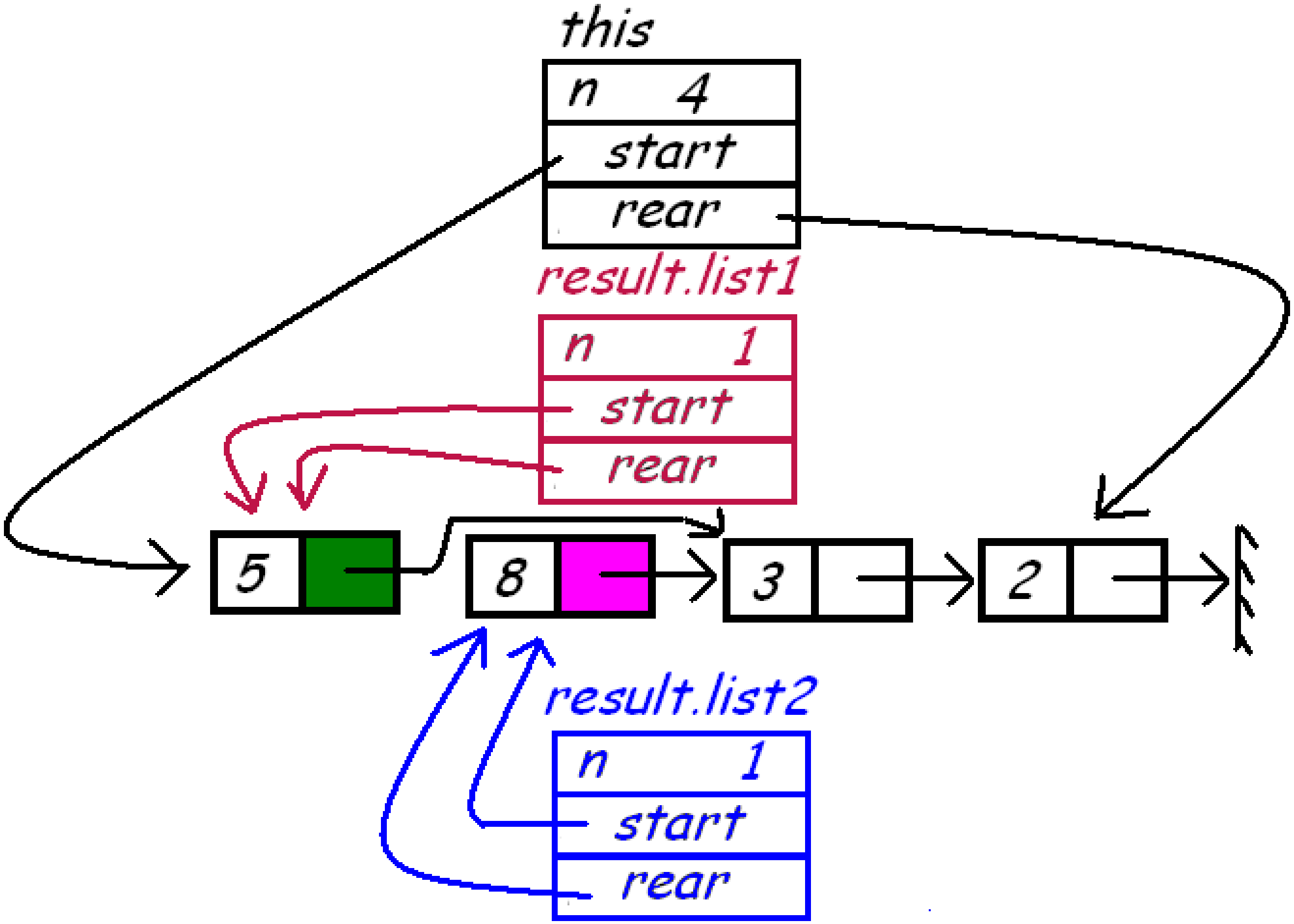
```
result.list1.rear = start;
```



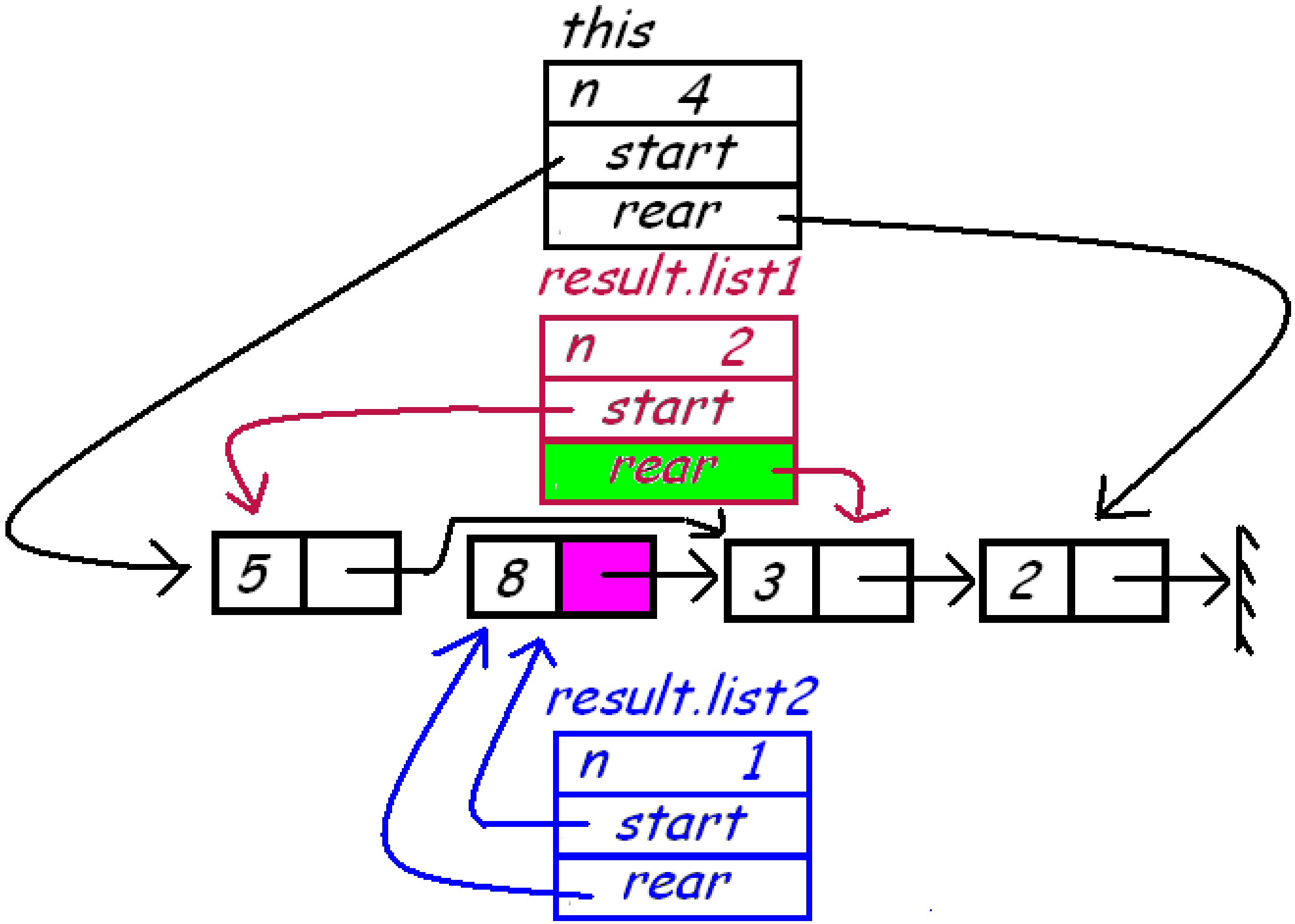
```
result.list1.n++;
```



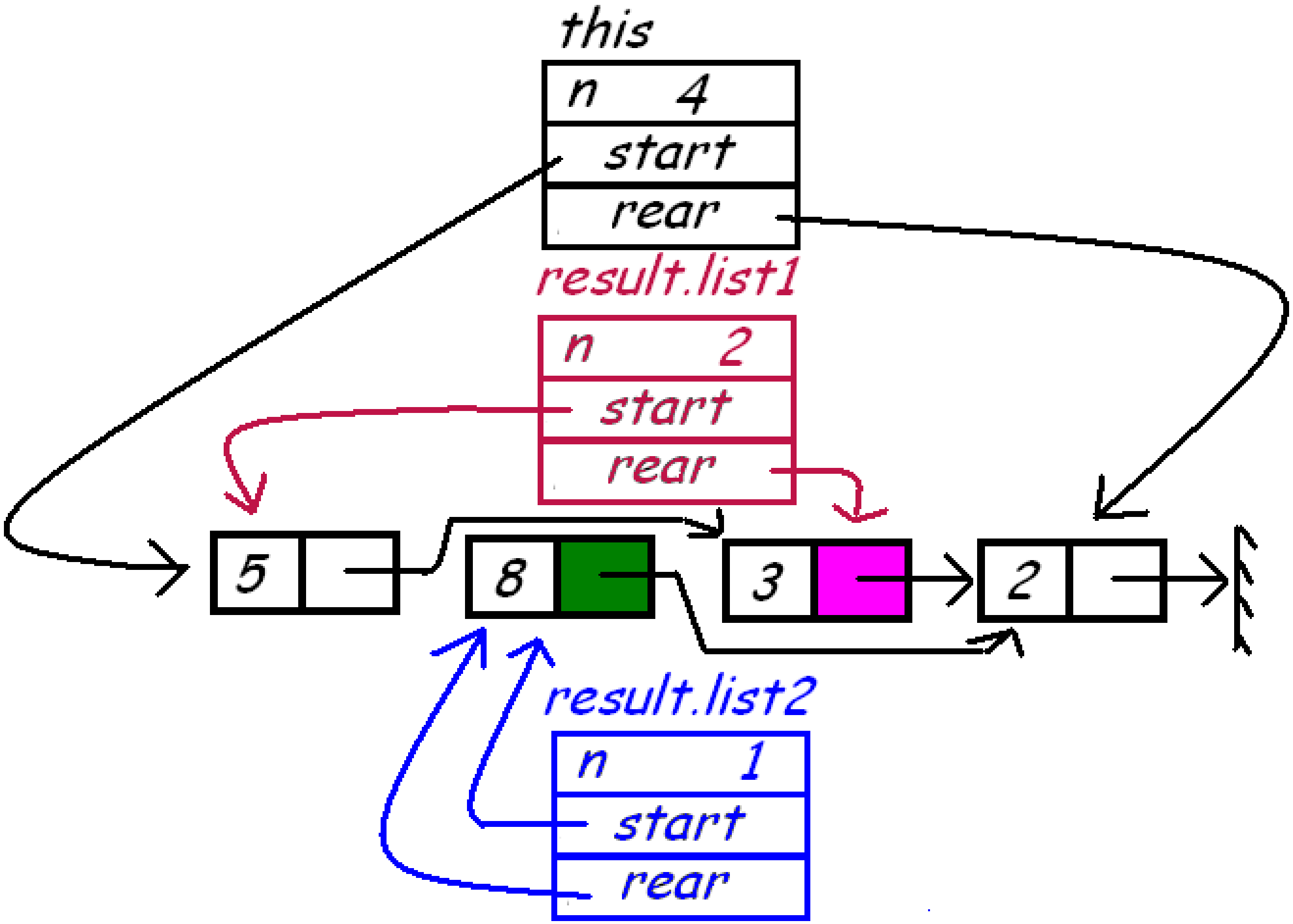
```
result.list2.start= start.next;
result.list2.rear = start.next;
result.list2.n++;
```



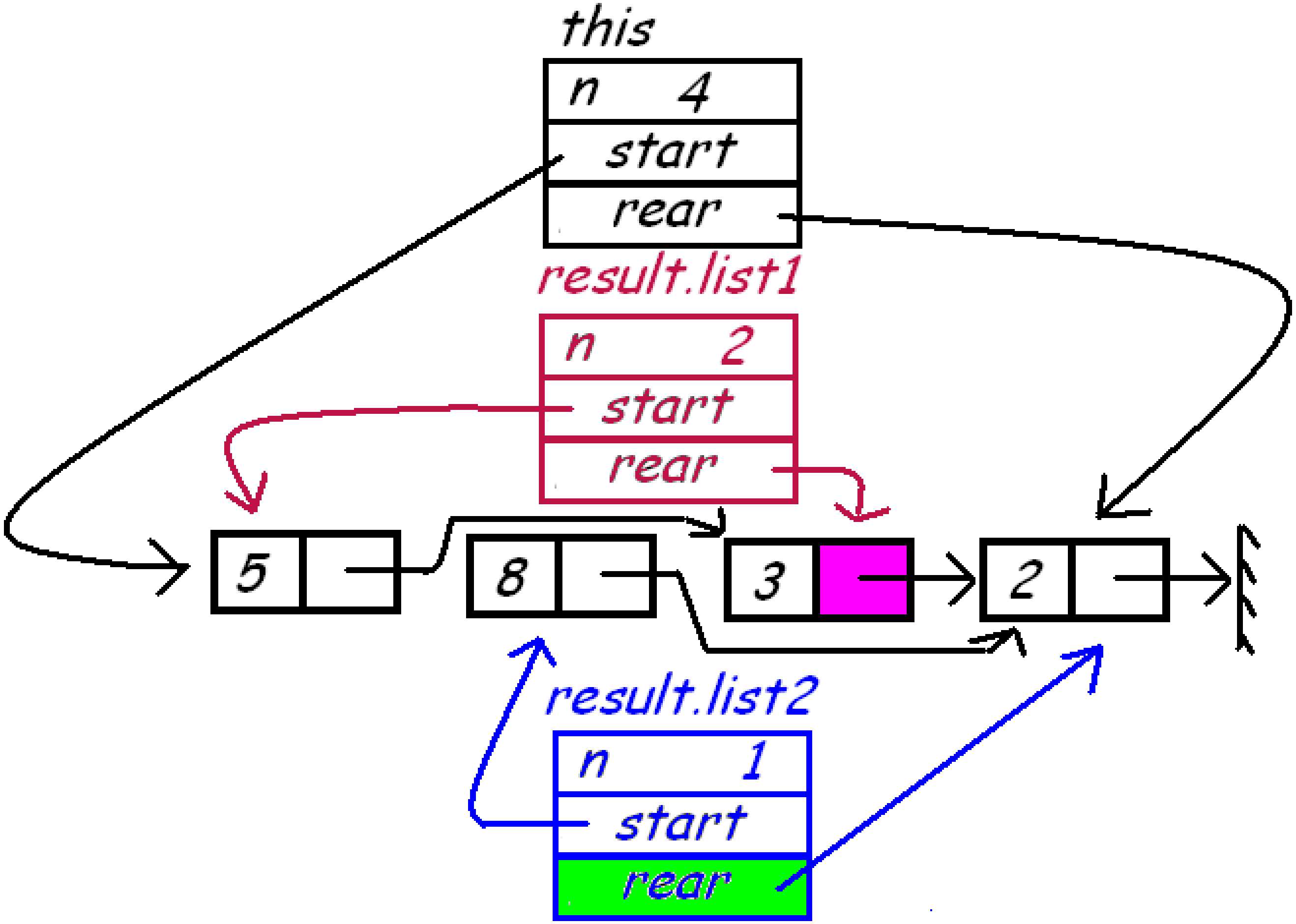
result.list1.rear.next= result.list2.rear.next;



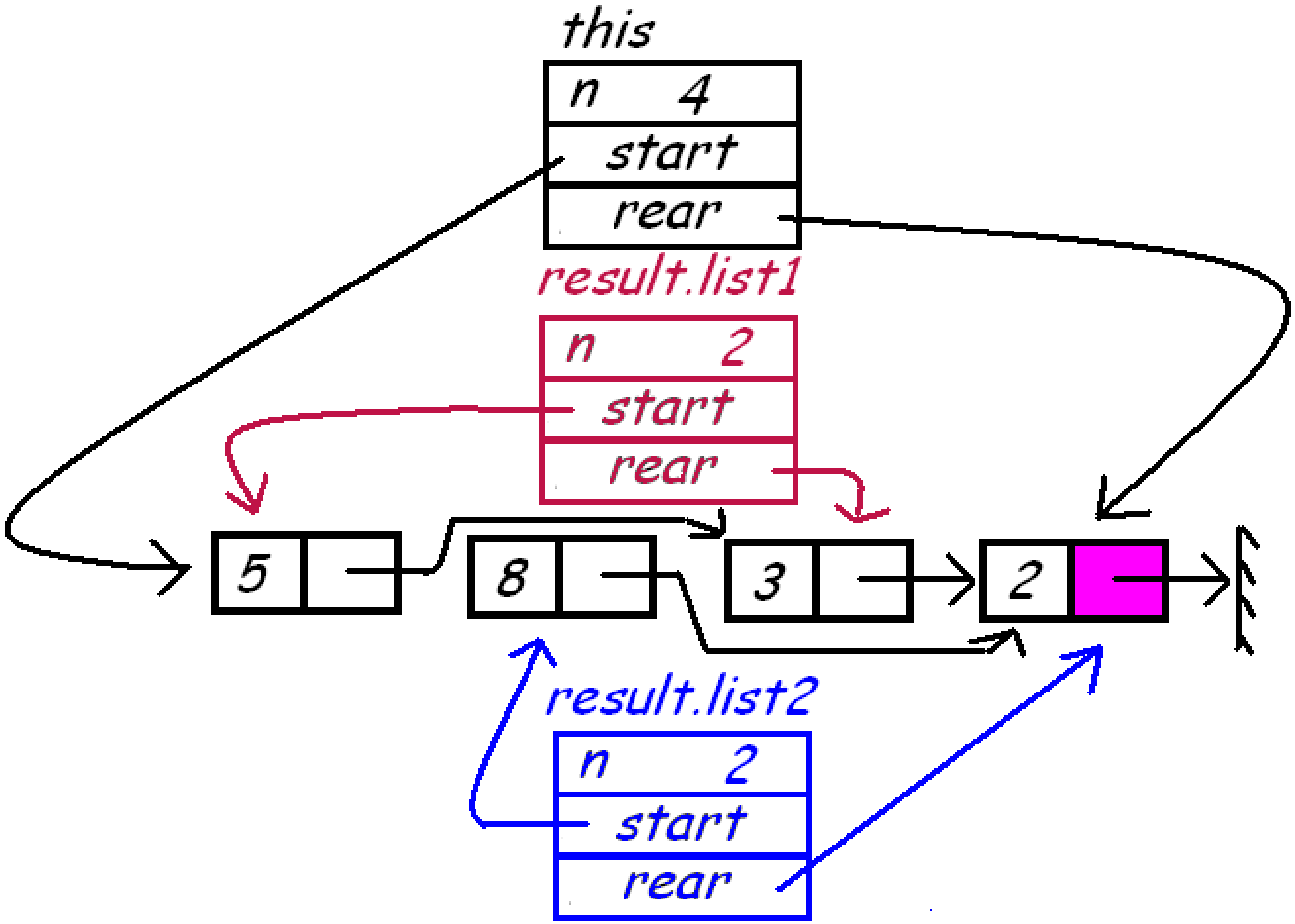
```
result.list1.rear= result.list2.rear.next;
result.list1.n++;
```



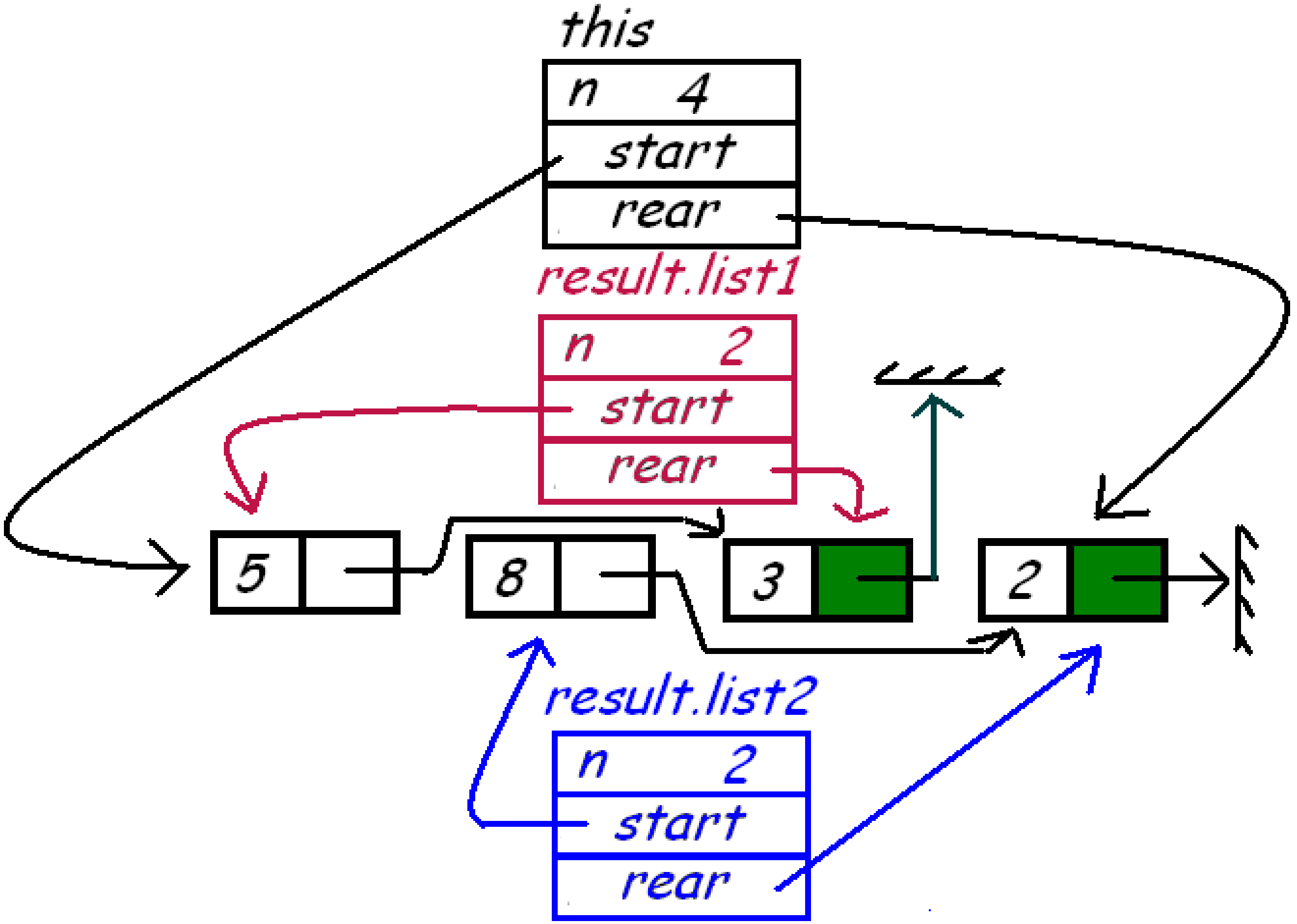
result.list2.rear.next= result.list1.rear.next;



result.list2.rear= result.list1.rear.next;

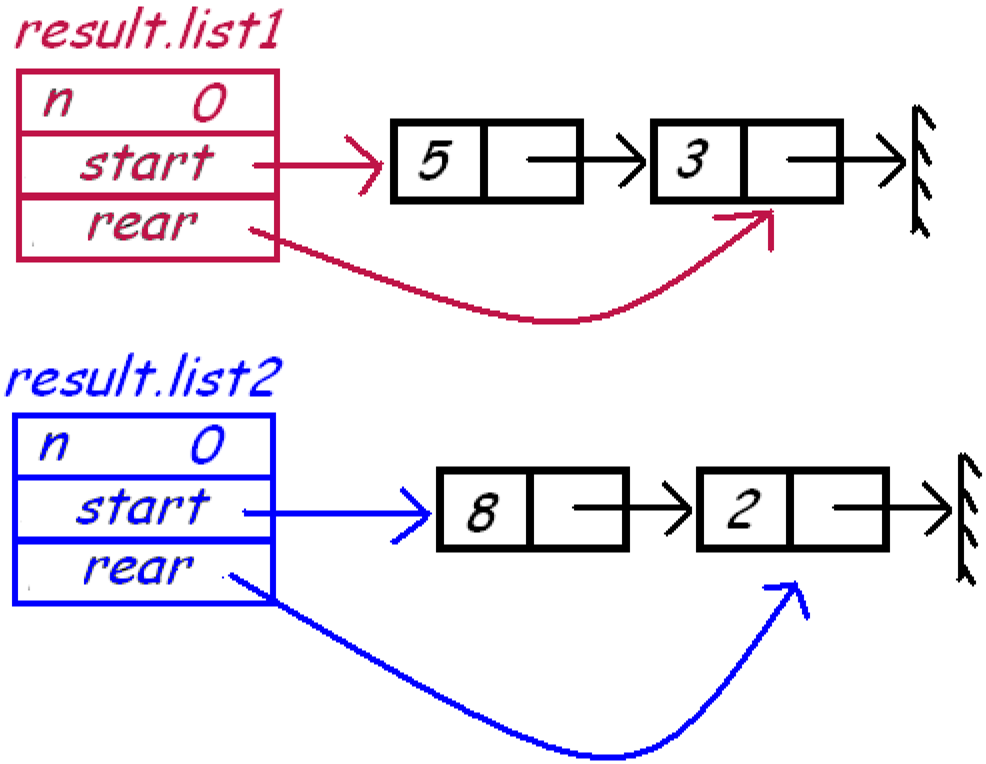


result.list2.n++;
The while loop now terminates:
while (result.list2.rear.next != null)



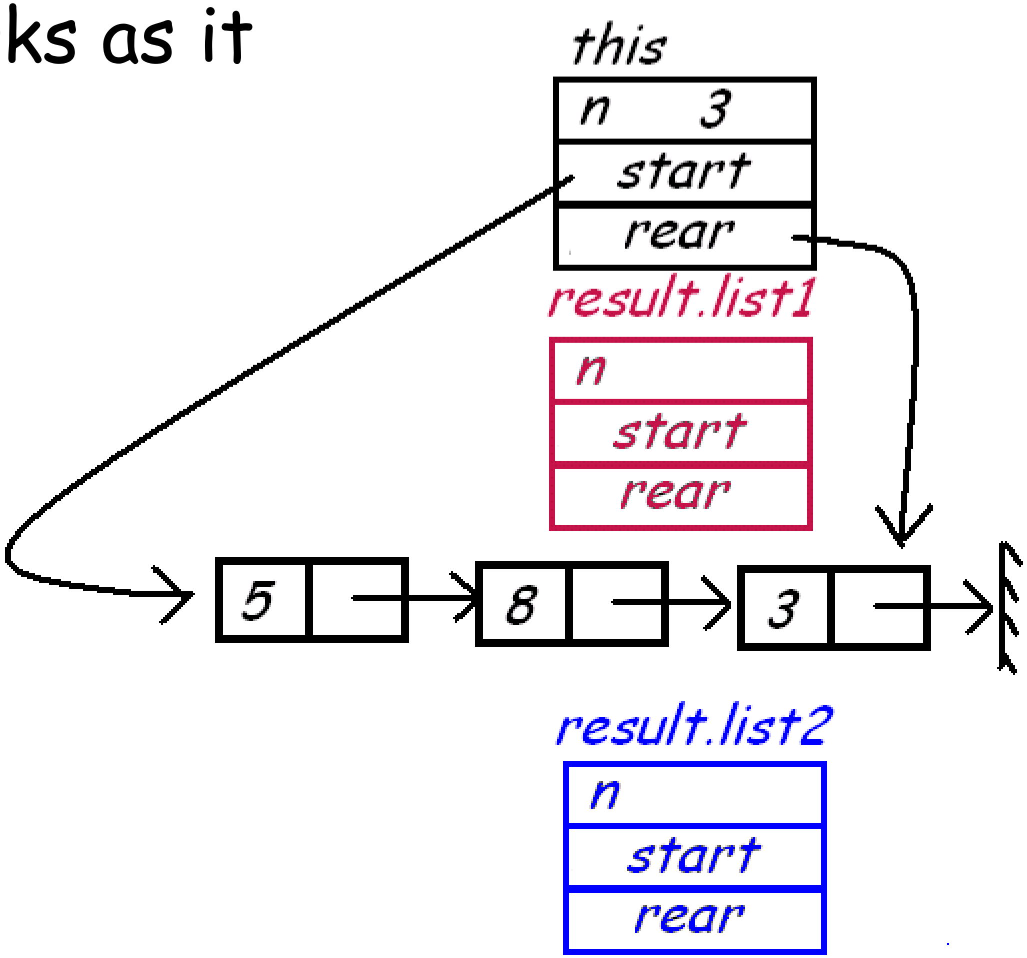
```
result.list1.rear.next= null;
result.list2.rear.next= null;
```

Redrawing the picture:



return(result);

The code I gave you is buggy. Try this example to see what goes wrong. Fix the program so it works as it should.


this
n 3
start
rear
result.list1
n
start
rear
5
8
3
result.list2
n
start
rear