Stack<Position> st =
new ListStack<Position>();

st.push(new Position(0,0));
Position p = st.pop();

Stack<Trail> searcher;
searcher.push();

stack LIFO DFS
top
Queue FIFO
front
back

public void add(E e);
public E remove();
public E element();
public int size();
public boolean isEmpty();

Array
ListNodes

Add (to back)
create new node with data
back's next is new node
back is back's next

if Empty
New Node with data
front is new node
back is new node

Remove
EmptyStack
save value of front
front is now front's next
return saved value

element
EmptyStack
return value of front

Size

```
for (int i=0; i<20; i++)
```

Init Continue Inc

```
int count=0;
for (ListNode temp = front;
  temp != null;
  temp = temp.getNext()){
    count++;
    temp.getValue();
}
return count;
```