Recurring Section 2

Aniruth – 5 PM

Announcements

- Welcome to our new recurring section students!
  - Welcome Form: https://forms.gle/bZqTGBUHo4xBrvLn9
- If you didn’t go to small group tutoring, check out:
  - https://edstem.org/us/courses/3834/discussion/500632
- Project 1: Deques will be due very soon
  - Start early! Do small pieces and build up.
  - Think about how you can use methods that you’ve already made to make life easier.
    - Start with the fundamentals.
  - Write tests as you go along. Make sure you fix all the small bugs before they accumulate.

Content Review

What IntLists are for

- Are arrays primitive or reference type? What about IntLists?
- Arrays are an indexed sequence of elements with fixed length.
  - The computer allocates a certain space within memory for this array; it's trying to have a variety of different arrays.
- Therefore, you must declare the length of an array during initialization.
- But that's not optimal! Why can't arrays just behave like they did in 61A so they can grow/shrink?
- That is a list, not an array. That's the whole point of the IntList – it changes in size, but it's harder to access values.
- The next slide is going to provide intuition behind these concepts as well as speed to make it less handwavy.

Demo

- What follows is not precisely correct – you don’t need to worry much about it as some of the specifics are very much out of scope.
- I want you all to focus on two things:
  - The procedures in creating the array/IntList objects
  - The time it takes to be able to access items within them
  - The ease of creating them
- Don’t worry about the specifics! This is only meant to provide some context for why you’re making lists at all.

Your Variables (Pointers):

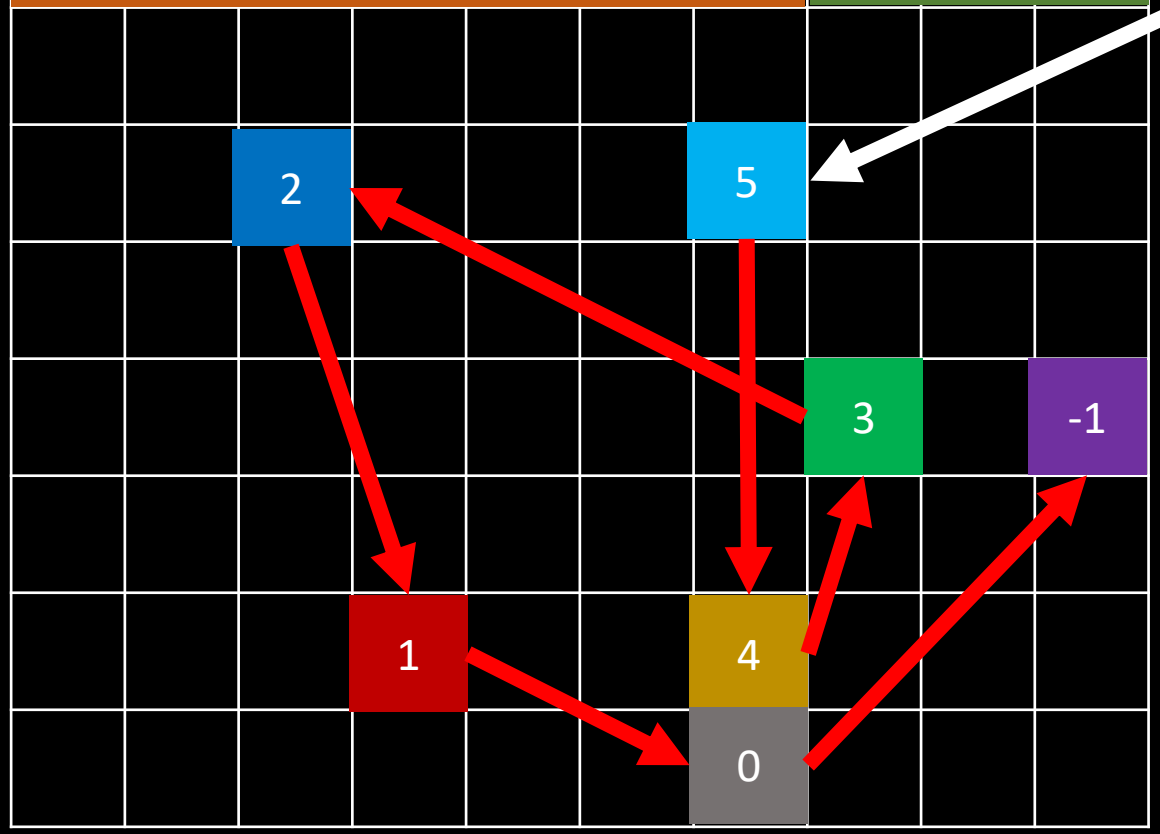
int[] example1 = new int[7];

int[] ex2 = new int[3];

[5,4,3,2,1,0,-1]

[5,2,4]

Your Computer:



5

IntList list1 = new IntList(5, null);

4

list1.next = new IntList(4, null);

3

list1.next.next = new IntList(3, null);

2

1

0

-1