CSE 12 – Basic Data Structures and Object-Oriented Design
Lecture 22

Greg Miranda & Paul Cao, Winter 2021

This lecture is being recorded

Announcements

- Quiz 22 due Friday @ 8am
- Survey 9 due Friday @ 11:59pm
- PA8 due Thursday, March 11 @ 11:59pm

Topics

- Streams
- Questions on Lecture 22?

Stream

Normally used to connect source and your program

source → input stream → program → output stream → destination

Why do we need streams?

Stream vs array

- Streams provide means for our program to input or output data
- What is the major difference between a stream and an array

A. stream is faster than arrays
B. stream uses less space than arrays
C. stream doesn’t allow random access while arrays can
D. None of the above

Memory Stream

- A stream that stores input or output in a buffer in memory
  - No random access

```
public void write(E data) {

  this.contents[this.back]= data;
  this.back++;
}

public E next() {
  E temp = this.contents[this.front];
  this.front++;
  return temp;
}
```

Initially, front and back are both 0

Whose index is front?

A. the element to be read
B. the location to be written into
C. A and B can be both be true under certain situation

Memory Stream

- A stream that stores input or output in a buffer in memory
  - No random access

```
public void write(E data) {

  this.contents[this.back]= data;
  this.back++;
}

public E next() {
  E temp = this.contents[this.front];
  this.front++;
  return temp;
}
```

Initially, front and back are both 0 and the capacity is 5.

Which of the following front/back pairs will result in a buffer overflow for write?

A. 3/4
B. 4/5
C. 3/5
D. More than one of them
E. None of the them

Memory Stream

- A stream that stores input or output in a buffer in memory
  - No random access

```
public void write(E data) {

  this.contents[this.back]= data;
  this.back++;
}

public E next() {
  E temp = this.contents[this.front];
  this.front++;
  return temp;
}
```

Initially, front and back are both 0 and the capacity is 5.

Which of the following front/back pairs will result in a buffer underflow for next?

A. 4/4
B. 4/3
C. 2/2
D. More than one of them
E. None of the them

Memory Stream

- A stream that stores input or output in a buffer in memory
  - No random access

```
public void write(E data) {

  this.contents[this.back]= data;
  this.back++;
}

public E next() {
  E temp = this.contents[this.front];
  this.front++;
  return temp;
}
```

What data structure is our memory stream similar to?

A. stack
B. queue
C. linked list
D. BST
E. None of the above

MemoryStream

- Convert array to circular array list
- How does this impact our implementation of the memory stream