CS62

Checkpoint 2 Review

Checkpoint logistics

In **lab** on Wednesday next week (11/4)

May bring one (double-sided) page of notes

Covers material from 9/24 (linked lists) through 10/22 (hashtables)

Resources: slides, quizzes, sample problems (a couple here a few more on the course webpage)

Topics

Linked lists:

- Variations: singly linked, doubly linked, tail reference, circularly linked
- Characteristics
  - Run-time of methods
  - Advantages of different variants
- Implementation/coding
  - Review LinkedList class (and Node class)
  - Should be comfortable writing/reading additional methods

Topics

Stacks and queue

- How they work
- Different implementations
  - ArrayList
  - Linked list (singly, doubly, tail reference)
  - Characteristics: run-time
- Use cases

Topics

Sorting:

- Algorithms: selection, insertion, mergesort, and quicksort
- Characteristics
  - How they work and what properties they maintain
  - Run-time (worse case, best case, average case)
  - Stable
  - In-place
- Pros/cons of different algorithms and when to use

Topics

Hashtables

- Why collisions occur
- Collision resolution techniques: chaining, open addressing
  - How they work
  - Run-time (best case, worst case, average case)
  - Pros/cons of two approaches
- Open addressing: probing
  - linear probing (should be able to do on examples)
  - quadratic probing, double hashing (know what they are and benefits)
- Hash function
  - purpose
  - properties (deterministic, fast, approximate simple uniform hasing)
  - approaches: division method, multiplication method

Topics

Java

- New interfaces
  - Iterator interface
  - Iterable interface
  - Comparable interface
  - Comparator interface
- What they’re used for
- How to implement

Lab today

In groups of 2 or 3, come up with at least 2 or 3 potential checkpoint 2 questions *along with the solution*. At least one of the questions should be a coding question (either reading or writing code).

Put them into a file and send them to me via slack and I'll post them at:

https://pomona.box.com/s/s7nzrs4rblk5ihpbm6hlt0dknag2gwfq

More questions

- More problems and solutions on the course webpage

- https://home.cs.colorado.edu/~main/questions/chap12j.html
  - **Ignore** the following questions:
    - Short answer: 3, 11, 12
    - Multiple choice: 4, 5, 11, 12
  - **Modify** the following questions
    - Short answer: 8 (assumed 4 for the pivot),

A few additional practice problems

Run times

You are given the observed running times of six different programs based on different input sizes. On each row of the table below, indicate your best hypothesis for the order of growth of the program. Please choose among constant, linear, quadratic, cubic, and none of these (an answer can be used more than once).

Observed running time					*Order of growth hypothesis*
Program	**1 million inputs**	**2 million inputs**	**4 million inputs**	**8 million inputs**	**Constant / linear / quadratic / cubic / none of these**
Program 1	15 min	1 hr	6 hr	2 days	
Program 2	0.1 sec	0.21 sec	0.45 sec	0.89 sec	
Program 3	2 min	16 min	2 hr	16 hr	
Program 4	30 min	4 hr	32 hr	5.3 days	
Program 5	3.02 days	3.01 days	3.02 days	3.01 days	

Iterators

A programmer would like to traverse an ArrayList in reverse order (from last element to first element). Modify the class ArrayList class we wrote in class to provide such an iterator.

```
public class ArrayList<E> implements List<E>, Iterable<E> {
     //instance variables data and size
     public Iterator<E> iterator() {
          return new ArrayListIterator();
     }
    private class ArrayListIterator implements Iterator<E> {
     //your implementation
     }
 }
```

More problems and solutions on the course webpage

Solutions

Solution: run times

You are given the observed running times of six different programs based on different input sizes. On each row of the table below, indicate your best hypothesis for the order of growth of the program. Please choose among constant, linear, quadratic, cubic, and none of these (an answer can be used more than once).

Observed running time					*Order of growth hypothesis*
Program	**1 million inputs**	**2 million inputs**	**4 million inputs**	**8 million inputs**	**Constant / linear / quadratic / cubic / none of these**
Program 1	15 min	1 hr	6 hr	2 days	none
Program 2	0.1 sec	0.21 sec	0.45 sec	0.89 sec	linear
Program 3	2 min	16 min	2 hr	16 hr	cubic
Program 4	30 min	4 hr	32 hr	5.3 days	none
Program 5	3.02 days	3.01 days	3.02 days	3.01 days	constant

Solution: Iterators

```
public class ArrayList<E> implements List<E>, Iterable<E> {
     //instance variables data and size
     public Iterator<E> iterator() {
          return new ArrayListIterator();
     }


    private class ArrayListIterator implements Iterator<E> {
        private int i = size -1;

        public boolean hasNext() {
            return i >= 0;
        }

        public E next() {
            return data[i--];
        }

        public void remove() {
        }

    }
 }
```