Welcome back!
Wake-up quiz – Heaps

- Is this a min-heap?
Wake-up quiz – Heaps

- Is this a min-heap?

- Yes
Week 7 class evaluation

- Overall, everything seems to be ok.
- Comments (slightly edited):
  - “Would there be a way for us to know what we got marked down on for previous programming assignments?”
    - Email.
  - “Could u talk about the HW assignment more detailed?”
    - Yes, today
Agenda for today

- Heaps and Priority queues
- Better design
- Assignment 4
- Assignment 5
Assignment 3 gotchas

• Hint number 8: Use a debugger
  – `cout << "I'm at the root!"`  
  – `gdb myProgram (in c++)`
    • If I show it in class, noone will remember
      – Start using it

• Hint number 9: Start early
  – It will probably never change :-)


Assignment 4 – part 1

- Implement a Binary Min-Heap
  - Use any language
- Use it as a Min-priority-queue
- Accept unspecified number of elements
  - Grow as necessary
    - Do not pre-store elements
Assignment 4 – part 1

- Heap implementation
  - Use an array to store elements
  - The elements have more data than just a key
    - Make a data structure for a heap node
    - Make a data structure for a heap that contains heap nodes
  - Support Min-Heap-Insert and Heap-Extract-Min
    - Get inspired by chapter 6
Better design

• Early term implementation

```c
struct HeapNode {
    int key;
    string data;
};
```

• This is fine for our purposes in this class but not how it is done in practice.
Better design

- What if, for example, I want to use a `double` as key instead of `int`?
- Use generic types for our data structures.
- All Java Collections classes use generic types. Examples:
  - `ArrayList<E>`
  - `LinkedList<E>`
  - `TreeSet<E>`
public class HeapNode<T extends Comparable<T>> {
    private T key;
    private String name;

    public HeapNode(T key, String name) {
        this.key = key;
        this.name = name;
    }
}
Better design

- What if I want to make several implementations of the same concept?
- Generalize the data structure
  - In Java, we would use interfaces
  - In C++, we would use templates
- For example, Queue<E> interface in Java has several implementations, including:
  - LinkedList<E>
  - SynchronousQueue<E>
public interface IMinHeap<T extends Comparable<T>> {
    public abstract void minHeapify(int i);

    public abstract void buildMinHeap();

    public abstract HeapNode<T> heapExtractMin();

    public abstract void minHeapInsert(HeapNode<T> x);

    public abstract void heapSort();
}
MinHeap implementation

```java
public class MinHeap<T extends Comparable<T>> implements IMinHeap<T> {

    private final int initialSize = 128;
    private HeapNode<T>[] elements;

    private int heapSize;

    public MinHeap() {
        elements = new HeapNode[initialSize];
        heapSize = 0;
    }
}
```
Assignment 4 – part 2

• Simple cryptographic system

• Principles of cryptography
  – The encryption/decryption *method* is known
  – The *cipher* is known
  – The *key* is unknown

• Cryptanalysis
  – Finding the *key* to decrypt the cipher
Assignment 4 – part 2

- Encryption
  - For every word $w$ in message $m$ with encryption key $k$
    - Select $k$ random words and attach a number in ascending order
    - Input $w$ and attach a number
  - Randomize the order of all words in the message

- This is how I produced hidden messages for you to decrypt.
Assignment 4 – part 2

• Decryption
  – For a hidden message $h$ with encryption key $k$ and a known number of words $w$
    • Extract the $k$ words with smallest number attached
    • Print the $(k+1)$th word
    • Repeat $w$ times

• This is what you should do
  – There are five hidden messages
  – $k$ is unknown but I give you hints.
Reading the key

- As usual, input should be possible from a file.
- The key is not part of this file
  - Use a parameter
- Remember:
  - “myProgram < f” redirects f to the standard input of myProgram.
  - Parameters have to come before this
C++ parameters

// argc: Number of arguments
// argv: Argument vector
int main(int argc, char * argv[]) {
    int secretKey;
    // argv[0] is the program's name
    istringstream keyStr(argv[1]);
    keyStr >> secretKey;
}

C++ parameters

- **Linux/Unix/Mac OS**
  - `g++ myProgram.cpp -o myProgram`
  - `./myProgram 1 < dwarves_key1`
    - Runs the program with the secret key 1, using the hidden message dwarves_key1

- **Windows**
  - `g++ myProgram.cpp -o myProgram`
  - `myProgram.exe 1 < dwarves_key1`
    - Runs the program with the secret key 1, using the hidden message dwarves_key1
// args is the arguments list
public static void main(String[] args) {
    int hiddenKey = Integer.parseInt(args[0]);
}
Java parameters

● **Linux/Unix/Mac OS**
  - `javac myProgram.java`
  - `java myProgram 1 < dwarves_key1`
    - Runs the program with the secret key 1, using the hidden message `dwarves_key1`

● **Windows**
  - `javac myProgram.java`
  - `java myProgram 1 < dwarves_key1`
    - Runs the program with the secret key 1, using the hidden message `dwarves_key1`
Java standard in

- By the way, in Java, there is no `cin` or `cout`
- We have to wrap something around `System.in`
- Scanner is a nice class for that

```java
Scanner scan = new Scanner(System.in);
int myInt = scan.nextInt();
String myStr = scan.next();
```
Assignment 5

• Optional
  – But only if you have more than 380 points!
• Due one week after assignment 4
• Implement quicksort
• Implement at least two other sorting algorithms
• Compare performance
  – Small write-up, for example with graphs
Programming languages

- We have looked at C++ and Java
- There are many programming languages
  - C++ and Java are practical and widely deployed
  - Some languages are impractical
- One of the goals of this course is to be able to learn new languages
- Concepts are often the same
Programming is fun

What is this?
Hello World in brainfuck

- *Brainfuck* is an impractical language
  - But it is theoretically as powerful as Java and C++
    - It is Turing complete
- Also called an esoteric programming language
- The point:
  - You can learn any language, but it takes practice
Thank you

Questions?