Announcements

• Complete Quiz 5 by tomorrow discussion section.
• Midterm on Thursday starts at **11:15am sharp**.
• In class, closed book, closed notes.
Review

• Array
• Big-O Notation
• Sorting
• Stacks
• Queues
• Linked Lists
• Recursion
Array

• Java code involving an array.
  – `For` loop over an array
  – Prelim test question (f), (g)

• Cost:

<table>
<thead>
<tr>
<th></th>
<th>Insertion</th>
<th>Search</th>
<th>Deletion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unordered</td>
<td>$O(1)$</td>
<td>$O(N)$</td>
<td>$O(N)$</td>
</tr>
<tr>
<td>Ordered</td>
<td>$O(N)$</td>
<td>$O(\log N)$</td>
<td>$O(N)$</td>
</tr>
</tbody>
</table>
Array

• Binary search in an ordered array.
• Loop-based version and recursive version.
• **Code** that implements the binary search.
• $O(\log N)$ cost.
• Elements must be ordered.
Big-O Notation

• Basic program analysis
• “Big-O” lecture slides example 1-6.
• Assignment 2 – written part
• In increasing order:
  \( O(1), O(\log N), O(N), O(N \log N), O(N^2), O(2^N), O(N!), O(N^N) \)
Big-O Notation

• Basic program analysis

• Example 1:

```c
int function1 (int n)
{
    int sum = 0;
    for (int i=n; i<100*n; i++)
    {
        for( int j=0; j<i; j=j+2 )
        {
            sum = sum + i*j*n;
        }
    }
    return sum;
}
```
Big-O Notation

• Basic program analysis

• Example 2:

```c
int function2 (int n)
{
    int sum = 0;

    while(n>=1)
    {
        sum = n*n*n;
        n = n/2;
    }
    return sum;
}
```
Big-O Notation

• Basic program analysis
• Example 3:

```c
int function4(int n) {
    int sum = 0;
    for (int i = 1; i < n; i = i*2)
        for (int j = 1; j < i; j = j*2)
            sum++;
    return sum;
}
```
• **Bubble sort** uses repeated comparisons and swaps to find the biggest element in each pass, and positions it towards the end of the array.

• **Selection sort** reduces the cost of swap by only performing one swap at the end of each pass.

• **Insertion sort** eliminates swaps and replaces them with copies, which are 3 times faster.
Sorting

• All three methods are comparison based.
• All of them have an average comparison cost of $O(N^2)$
• Code that implements bubble and selection sorting.
Stacks

• LIFO
• Push, Pop
  – Both $O(1)$ cost
• Code that implements push/pop
• Application – Delimiter matching
  – Left delimiters stored in stack
  – Right delimiter causes a stack pop and match
  – Popping an empty stack?
  – Stack non-empty after entire string processed?
Stacks

• Application – Postfix notation
  – Know how to evaluate a postfix notation
• Application – Program stacks
• Application – Backtracking (assignment 3)
Queues

• FIFO
• Insert (enqueue), Remove (dequeue)
  – Both O(1) cost
• Circular queue
• Code that implements insert/remove
• Priority Queue
  – Can be implemented using an ordered Array, or an ordered Linked List
  – Insert O(N), Remove O(1)
Linked List

- Consists of many link objects.
- Links are self-referential.
- A reference is a pointer (memory address), it does not contain the actual object.
- `first` variable.

Comparison of LinkedList vs. Array
- Storage, indexing, memory usage.
- Easiness of insertion and deletion.
- Dynamic size.
Linked List

• Comparison of LinkedList vs. Array
  – Storage, indexing, memory usage.
  – Easiness of insertion and deletion.
  – Dynamic size.

• Insert

• Search

• Delete
  – Code that implement the above methods
Linked List

• Double-ended List
• Sorted Linked List
• Doubly-Linked List
Recursion

• **Base case** is very important.
  – Stack overflow error if base case does not exist.
• Be able to write basic recursive methods.
• Examples: triangle number, factorial, etc.

• Be able to analyze the base case, and also write down the output of a recursive method.
Recursion

• Example 1:

```c
int callme(int n)
{
    if (n == 1)
        return 1;
    else
        return callme(n-1);
}
```
Recursion

- Example 2:

```c
int callme(int n)
{
    if (n == 1)
        return 0;
    else
        return 1+callme(n/2);
}
```
Assignment Review

• Assignment 1: Running average
• Assignment 2: Number guessing
• Assignment 3: N-Queens