CSE 250 Recitation

Mar 13 - Mar 17: Midterm Review

Bounds/Asymptotic Complexity

$$f(n) = 5n^2 \log^2(n) + 8n^2 + 2^{\log(20n)}$$

$$g(n) = 2n \log(n) + 3n + 9 \log(n \cdot 2^n)$$

- Provide the simplified tight lower-bound for f (n) with the appropriate asymptotic choice (O, Ω, or ϴ).
- Provide the simplified tight upper-bound for g(n) with the appropriate asymptotic choice (O, Ω, or ϴ).

Runtime Complexity

Assume that `arr` has already been instantiated as:

```
val n = Random.nextInt(100000)
val arr = new ArrayBuffer[Int]()
for(i <- 0 until n){ arr.append(elem = 0) }
```

What is the runtime complexity of:

```
arr.insert(idx = k, elem = 42)
```

Runtime Complexity

Assume that `arr` has already been instantiated as:

```
val n = Random.nextInt(100000)
val arr = new ArrayBuffer[Int]()
for(i <- 0 until n){ arr.append(elem = 0) }
```

What is the runtime complexity of:

```
val m = Random.nextInt(1000000)
for(i ← 0 until m) { arr.insert(idx = arr.length, elem = m) }
```

Stacks and Queues

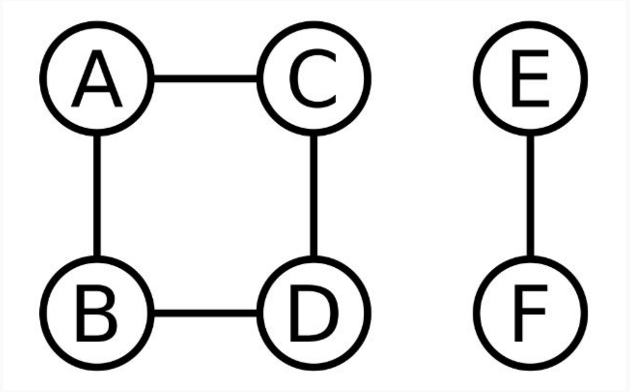
What does the following code print:

```scala
val seq1 = Seq(17, 73, 65, 0)
val seq2 = Seq(45, 1, 14, 48)
val queue = scala.collection.mutable.Queue()

for(i ← seq1){ queue.enqueue(i) }
for(i ← 0 until 3){ queue.dequeue() }
for(i ← seq2){ queue.enqueue(i) }
while(!queue.isEmpty){
  println(queue.dequeue())
}
```

Graphs

- How many vertices are in the largest connected component?
- List the vertices visited by BFSOne in the order in which they are visited. Assume edges are followed in alphabetical order of the opposite vertex.
- Do the same for DFSOne.



Graphs

As we discussed, the runtime of BFS is $O(|V| + |E|)$. To get to this runtime, we assumed the use of an adjacency list data structure. How does the runtime change if we use an edge list data structure?