Recitation 7 – BFS, Graphs and Exam I Debrief
Parallel and Sequential Data Structures and Algorithms, 15-210 (Fall 2014)
October 6th, 2014

1 Announcements

• How did the exam go?
• ThesaurusLab will be released soon!
• Questions from lecture?

2 Graphs

A undirected graph $G$ is a tuple of sets $G = (V, E)$, where

$V$ is a set of vertices (also known as nodes).

$E \subseteq \binom{V}{2}$ is a set of edges.

Do you recall the definition of a directed graph?

3 Graph Representations

1. Edge Set

A graph is represented by a set of edges. An edge $e = (x, y)$ is denoted by its endpoints. Notice that in an undirected graph, $(x, y)$ represents the same edge as $(y, x)$. This representation doesn’t provide an efficient means to access the neighbors of a vertex.

2. Adjacency Table

A graph is represented by a table with keys for all $v \in V$. $v \rightarrow S$, where $S$ is a set and any vertex $u \in S \iff (v, u) \in E$.

3. Array Sequences

This approach is slightly less general because it requires that the vertices be numbered from 0 to $n - 1$. For a given vertex $v$ with number $i$, the $i$th index of the sequence will contain the $S$ that $v$ would map to in the adjacency table representation.

We covered in class the costs of doing various operations with an adjacency table. What would be the cost of doing them with array sequences?
work          span

isEdge(G,(u,v))
map over all edges
map over neighbors of v
\(d_G^+(v)\)

4 Friends of Friends

Suppose we have just founded a new social networking site and naturally we have represented the network of friends as a graph using the adjacency table representation.

Suppose we want to find friends of our friends. These would be the neighbors of our neighbors, or all vertices that have a distance of 2 from a source vertex \(v\). We may use either of the following TABLE functions:

```ocaml
val tabulate : (key -> 'a) -> set -> table
val extract : ('a table * set) -> 'a table
```

Write the function \(\text{FoF}\) to solve this problem.

```ocaml
1 type graph = Set.set Table.table
2 fun FoF (G:graph) v =
3   let
4   in
5   in
6   end
```
5 Parallelism in BFS?

Q: Can BFS recursive calls be made in parallel?

Consider the following implementation of BFS:

```haskell
fun BFS G X v =
  let
    N = difference(find G v, X)
    X' = union(X, N)
  in
    reduce union Set.singleton(v) (map (BFS G X') (toSeq N))
  end
```

Q: Draw a graph that demonstrates the problem with this BFS implementation.

Q: How many times do nodes get visited?

Consider a complete graph of $n$ nodes. Let’s add one more node, called $v$, to this graph by connecting it to $k$ other nodes. Run our BFS starting at $v$. How many times do nodes get visited? Compare this to lecture’s BFS. What happens if $k = 1$? $\frac{n}{2}$? $n$?

Bonus: What is a general Big-O bound on this algorithm?
6 Coding BFS

Write a function to find the **diameter** of a graph using **breadth-first** traversal. Recall that the diameter of a graph is the length of the longest distance between two vertices in the graph.

To get started, first consider: how does this relate to BFS? What would be a good graph representation to use?

You may also want to write a helper function to get the neighbors of a set of vertices.

**Bonus:** Can you do better?