Recitation 8 – *Graphs and BFS*

Parallel and Sequential Data Structures and Algorithms, 15-210 (Spring 2015)

March 3rd, 2015

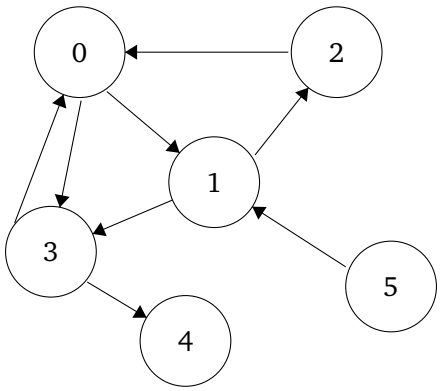
1 Announcements

- How was *Babblelab*?
- *RangeLab* has been released (due wednesday after spring break)!

2 Graph Representations

2.1 Yo What's my Rep(resentation)?

Consider the following graph.



Q: Construct the edge set representation of this graph:

A:

Q: Construct the adjacency table representation of this graph:

A:

Q: Construct the adjacency sequence representation of this graph:

A:

Q: Name any interesting properties of this graph (i.e. diameter, connectedness, special paths...)

A:

Q: Draw the BFS tree of this graph, starting with 0 as the source. Looking only at the BFS tree, how can we tell that there wasn't an edge from 0 to 2 in the original graph?

A:

2.2 Cost Bounds

Q: Fill in the costs of doing these graph operations for the adjacency sequence and adjacency table representations of a graph

	adj. table		**adj. seq**	
	work	*span*	*work*	*span*
`isEdge`$(G,(u,v))$				
`map over all edges`				
`map over neighbors of` v				
$d_G^+(v)$				

3 Friends of Friends

Q: Suppose we have just founded a new social networking site. Naturally we have represented the network of friends as a graph using the adjacency table representation.
Suppose we want to find all 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:

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

```
1  type graph = Set.set Table.table
2  fun FoF (G : graph) (v : key) : set =
3  let
4
5
6
7  in
8
9  end
```

4 Parellelism in BFS?

Consider the following implementation of BFS:

```
1  fun BFS G X v =
2  let
3   val N = difference (find G v,X)
4   val X' = union(X,N)
5  in
6   reduce union {v} (map (BFS G X') (toSeq N))
7  end
```

Q: Is this implementation correct?

A:

Q: Is this implementation efficient?

A:

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

A:

5 More BFS

Q: Describe an algorithm using BFS to find all nodes who are more than k hops away from the source vertex (work and span should not exceed that of BFS).

A:

Q: Describe an algorithm using BFS to find all nodes who are more than k hops away from every node within some subset $U \subset V$.

A:

Q: Describe an algorithm using BFS to find the diameter (the length of the longest shortest path) of a graph. What are the work and span of your algorithm?

A: