


```
fn components(edges: Stream<Edge>) -> Stream<Label> {  
    // convert each edge into a label.  
    edges.map(|(src, dst)| (src, src))  
        .iterate(|labels|  
            // join labels with edges, add old labels, take min.  
            labels.join(edges, |src, lbl, dst| (dst, lbl))  
                .concat(labels)  
                .argmin()  
        )  
}
```

Differential Dataflow

Echidnatron% cargo run --release --example bfs

Compiling differential-dataflow v0.0.1 (file:///Users/mcsherry/Projects/differential-dataflow)

Running `target/release/examples/bfs`

performing BFS on 50000000 nodes, 100000000 edges:

loaded; elapsed: 4.606321882922202s

wave 0: avg 0.00008189163205679506

wave 1: avg 0.00008574895199853927

wave 2: avg 0.00017252071597613394

wave 3: avg 0.00007136030797846616

wave 4: avg 0.00008547972002997995

wave 5: avg 0.00009294764802325516

wave 6: avg 0.00009670707699842751

wave 7: avg 0.00010349303390830755

wave 8: avg 0.00008258083905093372

wave 9: avg 0.00008172653894871473

wave 10: avg 0.00010690367000643165

wave 11: avg 0.00016434095602016897

wave 12: avg 0.00009046116506215185

wave 13: avg 0.0001997114790137857

wave 14: avg 0.00014574242196977138

wave 15: avg 0.00007636803097557276

wave 16: avg 0.00009653931495267897

wave 17: avg 0.00011602789699099958

wave 18: avg 0.00009280436299741269

wave 19: avg 0.00008407771203201264

wave 20: avg 0.00009858175099361687