


```
let nodes = /* pairs (node, bool) */;  
let edges = /* pairs (node, node) */;
```

```
for t in times {  
    nodes.insert(..); nodes.remove(..);  
    edges.insert(..); edges.remove(..);  
}
```

```
nodes.join(edges)           // one hop neighbors
    .concat(nodes)          // plus original nodes
    .distinct()              // extended neighborhood
```


input streams of changes

“program” : dataflow assembly

dataflow execution


```
let nodes = /* pairs (node, bool) */;  
let edges = /* pairs (node, node) */;
```

```
nodes.join(edges)           // one hop neighbors  
    .concat(nodes)         // plus original nodes  
    .distinct()             // extended neighborhood
```

```
for t in times {  
    nodes.insert(..);  
    edges.insert(..); edges.remove(..);  
}
```

```
let nodes = /* pairs (node, bool) */;  
let edges = /* pairs (node, node) */;
```

```
nodes.join(edges)      // one hop neighbors  
    .concat(nodes)     // plus original nodes  
    .distinct()        // extended neighborhood
```

```
for t in times {  
    nodes.insert(..);  
    edges.insert(..);  
}
```