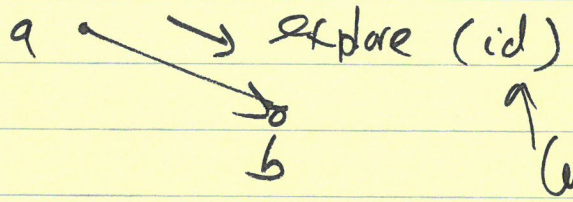
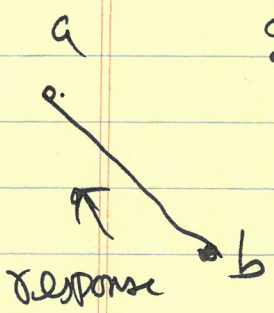
9-10-19

Floodmax - termination





Get a response
(i) reject (immediately, or next round)
(ii) ok, a done message.

Node a is a leaf if all explore() messages sent by a get a reject as the response.

If a node is a leaf, send a Completed() message to the parent

If a node is a non-leaf,
{ wait till all explore() messages get a response each:
(reject / Completed)
Send a Completed() message to the parent
}

When a node (or a process) v receives a reply for each explore() message it had sent & if no explore() message with id larger than v has been received, the leader & it is v knows it is time to terminate

Exploration: leader → all (go down & create the tree)

Replies → Convergecast (go up)

Termination → go down the tree

Breadth First Search

i_0 : root of the tree.

Build BFS tree rooted at i_0.

{ uses: broadcast a message, send a message from i_0 to another Process.

Shortest path

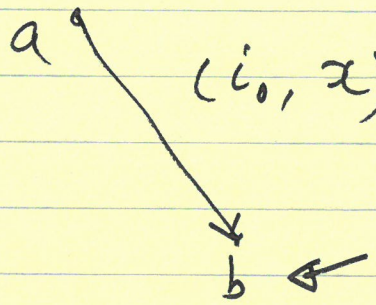
Smallest # of vertices

}

Initially, i_0 sends a message containing $(i_0, 0)$ to all neighbors

i_0: root id

0: # of hops between root & sender of this message.



if b is already part of the BFS tree rooted at i_0 { ignore this message }

else { if this same message is received from >1 neighbor in this round {

Choose one of them to be the parent. // Joined the tree

hopcount to the root = x+1

forward message $(i_0, x+1)$ to all neighbors as needed

}

else { Join the tree;

forward $(i_0, x+1)$ to all neighbors as needed }

Time: $O(diam)$

messages: $O(|E|)$

(4)

We have a tree rooted at say x. (T_x)

Each process has a value, known only to itself (local value)
Want to find the max_value of all local values.
x broadcasts a message (saying max_value is to be found)
On receiving this message {

Using convergecast

if i'm a leaf, {send my local value to parent}
else {
wait for a message containing a value from each child.
choose largest among values received from all children and its own local value & send this chosen value to parent
}

messages: $2.(n-1)$
Time: $2.$ (depth of tree)

min, sum, mat, average. Can be done this way.

Read about BFS tree in ~~[illegible]~~ directed graphs. [not all edges / links are bidirectional] from the book

Shortest paths.

Shortest paths tree.

i_0 = root.

all links have a non-negative costs.

each

cost

Bellman Ford algorithm

$dist_x$ = distance that x is from i_0.

initially, $dist_x = \infty$ if $x \neq i_0$

&

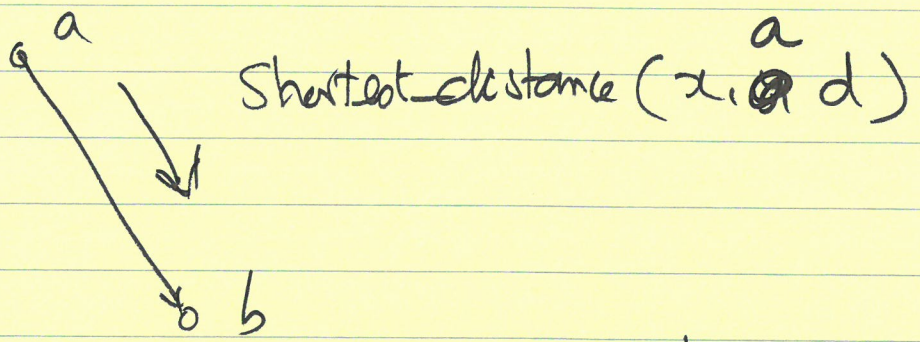
$dist_x = 0$ if $x = i_0$

i_0 sends ~~[illegible]~~ shortest_dist_ $(i_0, i_0, 0)$ to all neighbors.

root of tree

sender of this message

Shortest distance from root to sender of this message as of now.



b receives message Shortest_distance(x, a, d)

b performs "relaxation" step

if $d + cost(a,b) < dist_b$ (for B shortest path tree ~~out~~ rooted at x) {

// this message is "useful"

$dist_b$ (sp tree rooted at x) = $d + cost(a,b)$;

parent(b) = a;

forward dist(x, b,) to all other neighbors

}

else { discard message }

Time:

messages: