6.S899 Distributed Graph Algorithms Lecture 6 MIT 10/17/2014

Material covered:

Two-party communication lower bound of the equality function:
Follows section 11.3 in “Roger Wattenhofer, notes for Lecture 11 “Hard Problems” (co-authored with Stephan Holzer) of the lecture Principles of Distributed Computing taught at ETH Zurich.
http://dcg.ethz.ch/lectures/podc_allstars/lecture/chapter11.pdf

Lower bound for Spanning Tree verification and Approximation of MST that matches the runtime of the exact MST-algorithm from lecture 5:
Simplified version (using an assumption) of the proof in “Atish Das Sarma, Stephan Holzer, Liah Kor, Amos Korman, Danupon Nanongkai, Gopal Pandurangan, David Peleg and Roger Wattenhofer, Distributed Verification and Hardness of Distributed Approximation, SIAM Journal on Computing volume 41, number 5 (special issue of STOC 2011), pages 1235-1265, 2012.”

LAST LECTURE:

- □ MST IN $\tilde{O}(\sqrt{n} + D)$ VIA $(\sqrt{n}, \sqrt{n} \log n)$-FRAGMENTS

TODAY:

- □ $\Omega(k)$ FOR TWO-PARTY COMMUNICATION COMPLEXITY OF EQUALITY VERIFICATION
- □ $\tilde{\Omega}(\sqrt{n} + D)$ FOR SPANNING TREE VERIFICATION
- □ $\tilde{\Omega}(\sqrt{n} + D)$ FOR poly(n)-APPROXIMATION OF MST
- □ $\tilde{\Omega}(\sqrt{n} + D)$ FOR poly(n)- APPROXIMATION OF (WEIGHTED) SINGLE SOURCE SHORTEST PATHS (SSSP)

RESTATE: CONGEST MODEL / MST

DEF: TWO-PARTY COMMUNICATION COMPLEXITY

$$A \xleftrightarrow{\text{1 BIT / ROUND}} B$$

INPUT: $X \in \{0,1\}^k = (x_1, \dots, x_k)$ $Y \in \{0,1\}^k = (y_1, \dots, y_k)$

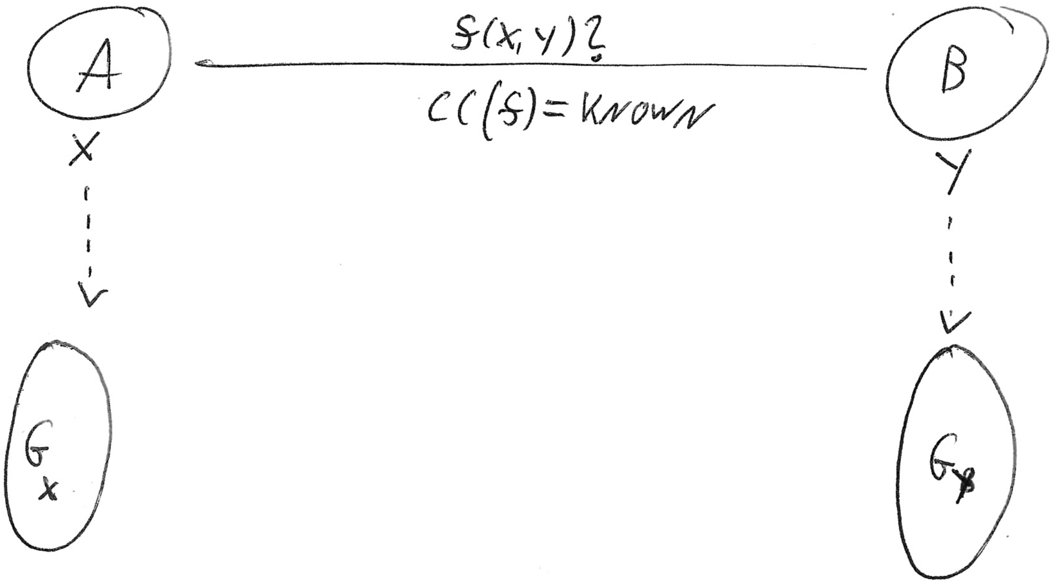
COMPUTE: $f(X, Y)$, $f : \{0,1\}^k \times \{0,1\}^k \longmapsto \underline{\underline{\{0,1\}}}$

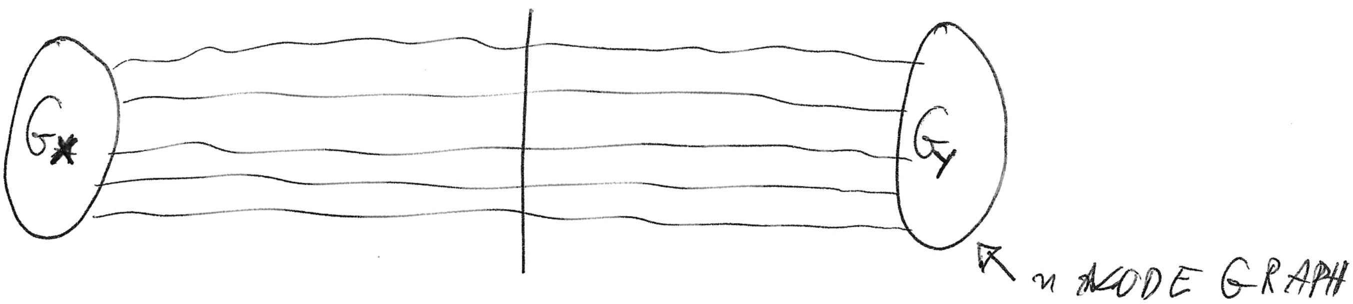
$CC(f) :=$ # BITS EXCHANGED BY THE BEST PROTOCOL IN THE WORST CASE.

IDEA:
- PROVE A TWO-PARTY LOWER BOUND
- TRANSFER IT INTO A MORE DISTRIBUTED SETTING.

NOTE: $\tilde{\Omega}()$ AND $\tilde{O}()$ IGNORE polylog(n)-FACTORS

GENERAL FRAMEWORK:





- TAKES $\Omega(g(k))$ ROUNDS TO EXCHANGE $CC(f)$ BIT.
- GRAPH HAS CERTAIN PROPERTY P IFF $f(X,Y)=1$.

$\Rightarrow$

$\Rightarrow$ VERIFYING THE PROPERTY TAKES $\Omega(g(k))$ ROUNDS.

EXAMPLE:

- P = "SUBGRAPH H_{XY} INDUCED BY MARKED EDGES (RED) IS SPANNING TREE OF G_{XY}"
- $k=\sqrt{n}$, $g(\sqrt{n})=\tilde{\Omega}(\sqrt{n})$, $D=O(\log n)$
- $f=EQ$, $CC(f)=\Omega(k)$

$\Longrightarrow$ VERIFYING IF H_{XY} IS ST TAKES $\tilde{\Omega}(\sqrt{n})$ $(+D)$

DEF: EQUALITY VERIFICATION

$$EQ(X,Y) := \begin{cases} 1 & : X = Y \\ 0 & : X \neq Y \end{cases}$$

DEF: MATRIX REPRESENTING f

$$M^f_{X,Y} := f(X,Y)$$

EXAMPLE: FOR EQ IN CASE $k=3$

$$M^{EQ} := \begin{array}{c|cccccccc} EQ & 000 & 001 & 010 & 011 & 100 & 101 & 110 & 111 \; \leftarrow X \\ \hline 000 & 1 & & & & & & & \\ 001 & & 1 & & & & & & \\ 010 & & & 1 & & & & & \\ 011 & & & & 1 & & & & \\ 100 & & & & & 1 & & & \\ 101 & & & & & & 1 & & \\ 110 & & & & & & & 1 & \\ 111 & & & & & & & & 1 \\ \uparrow Y & & & & & & & & \end{array}$$

DEF: $R \subseteq \{0,1\}^k \times \{0,1\}^k$ IS (COMBINATORIAL) MONOCHROMATIC RECTANGLE IF

- $(X_1, Y_1) \in R$ AND $(X_2, Y_2) \in R$ IMPLIES $(X_1, Y_2) \in R$
- THERE IS A FIXED z S.T. $f(X,Y) = z$ FOR ALL $(X,Y) \in R$

EXAMPLE:
- $R_1 := \{011\} \times \{011\}$
- $R_2 = \{011, 100, 101, 110\} \times \{000, 001\}$
- $R_3 = \{000, 001, 101\} \times \{011, 100, 110, 111\}$
- $R_4 = \{000, 001\} \times \{000, 001\}$

$M^{EQ} :=$

EQ	000	001	010	011	100	101	110	111	$\leftarrow x$
000	1	0	0	0	0	0	0	0	
001	0	1	0	0	0	0	0	0	
010	0	0	1	0	0	0	0	0	
011	0	0	0	1	0	0	0	0	
100	0	0	0	0	1	0	0	0	
101	0	0	0	0	0	1	0	0	
110	0	0	0	0	0	0	1	0	
111	0	0	0	0	0	0	0	1	
$\uparrow y$									

R_4 R_1 R_2 R_3

R_1, R_2, R_3 ARE MONOCHROMATIC, R_4 IS NOT

NOTE: • WHEN EXCHANGING A BIT A AND B CAN ELIMINATE THE CORRESPONDING ROWS.

• CAN STOP WHEN A MONOCHROMATIC ~~TRI~~ RECTANGLE IS LEFT

• MAYBE A AND B CAN DO BETTER THAN SENDING THE GIVEN BITS ?

DEF: FOOLING SET

$S \subseteq \{0,1\}^{\ell} \times \{0,1\}^{\ell}$ FOOLS f IF THERE IS A FIXED z s.t.

• $f(x,y) = z$ FOR EACH $(x,y) \in S$

• FOR $(x_1,y_1) \neq (x_2,y_2)$ RECTANGLE $\{x_1,x_2\} \times \{y_1,y_2\}$ IS NOT MONOCHROMATIC
  • $f(x_1, y_2) \neq z$ (OR / AND)
  • $f(x_2, y_1) \neq z$

EXAMPLE: $S = \{(000,000), (001,001)\}$ FOOLS EQ.

LEM I.: S FOOLS f, THEN $CC(f) = \Omega(\log |S|)$

PROOF: ASSUME ALG ALWAYS NEEDS $t \lneq \log |S|$ ROUNDS TO COMPUTE f

$\Rightarrow |\{0,1\}|^t = 2^t < |S|$ ACTION PATTERNS (BIT SEQUENCES SENT)

$\Rightarrow \exists (x_1,y_1), (x_2,y_2) \in S$ s.t. ALG PRODUCES SAME PATTERN P

$\Rightarrow$ A PRODUCES P ON (x_1,y_2) AND (x_2,y_1) AS WELL

$\Rightarrow$ AFTER t ROUNDS A DOES NOT KNOW IF B's INPUT WAS y_1 OR y_2 AND B " " " A's " " x_1 OR x_2.

BY DEF. OF FOOLING SET: • $f(x_1,y_2) \neq f(x_1,y_2) \Rightarrow$ A DOES NOT KNOW THE SOLUTION

• $f(x_2,y_1) \neq f(x_1,y_1) \Rightarrow$ B DOES NOT KNOW THE SOLUTION

$\lightning$ ASSUMPTION

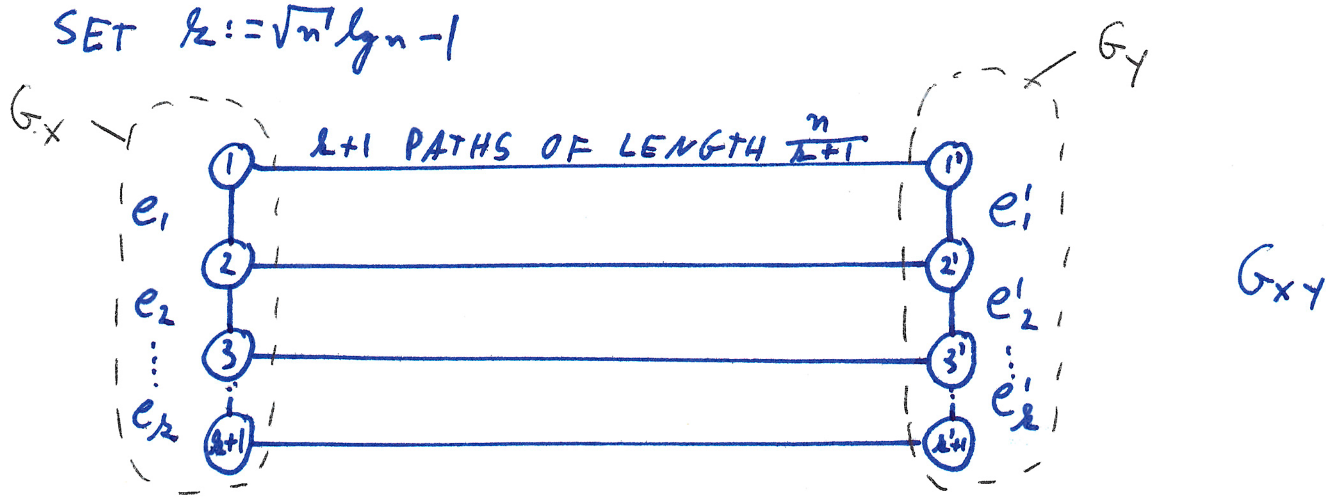
$\square$

THM: $CC(EQ) = \Omega(k)$

PROOF: $S := \{(x,x) \mid x \in \{0,1\}^k\}$ FOOLS EQ AND HAS SIZE 2^k. APPLY LEM. I. $\square$

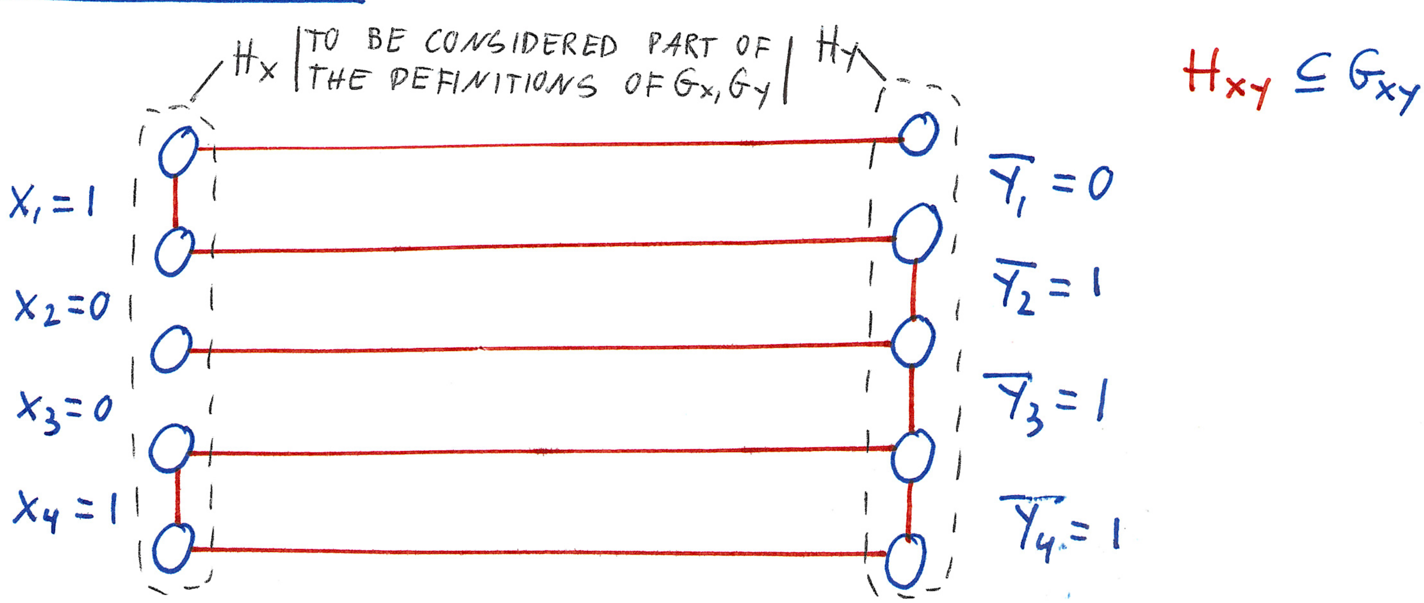
LEM II. ASSUME $x_i + \overline{y_i} \neq 0$ FOR ALL $1 \leq i \leq k$. DECIDING IF H_{xy} IS A SPANNING TREE OF G_{xy} TAKES $\tilde{\Omega}(\sqrt{n})$ ROUNDS IN THE FOLLOWING CONSTRUCTION.

SET $k := \sqrt{n} \lg n - 1$



H_{xy} CONSISTS OF ALL $k+1$ PATHS AND EDGES e_i s.t. $x_i = 1$ AS WELL AS EDGES e'_i IF $\overline{y_i} = 1$.

EXAMPLE FOR H_{xy}: $x = 1001$, $y = 1000$ $(k = 4)$



NOTE THAT H_{xy} IS NOT A ST OF G_{xy}.

PROOF OF LEM II:

1) H ALWAYS SPANS G:

$x_i + \overline{y_i} \neq 0 \Rightarrow$ THERE IS ALWAYS AN EDGE (ASSUMPTION) e_i OR e_i' ON AT LEAST ONE SIDE IN H_{xy}

2) H IS A TREE IF $x = y$

$x_i + \overline{y_i} = 1$ AS $x = y \Rightarrow$ ALWAYS EXACTLY ONE EDGE e_i OR e_i' IN H_{xy}.

3) G_x NEEDS $\tilde{\Omega}(\sqrt{n})$ ROUNDS TO EXCHANGE JUST ONE BIT WITH G_y.

AS WE CAN DECIDE IF $x = y$ WHEN WE CAN DECIDE IF H_{xy} IS SP ST OF G_{xy} (FOLLOWS FROM 1) AND 2)), ONE NEEDS TO EXCHANGE $CC(EQ) = \Omega(k) = \tilde{\Omega}(\sqrt{n})$ BIT.

$\square$

QUESTION: WHY IS THIS CHEATING?

- BECAUSE $D \approx \sqrt{n} / \log n$
  - $\rightarrow$ DON'T NEED EQ-LOWER BOUND
- NO USE OF "LIMITED BANDWIDTH" IN THE ARGUMENT, WORKS IN THE LOCAL MODEL AS WELL

SOLUTION?

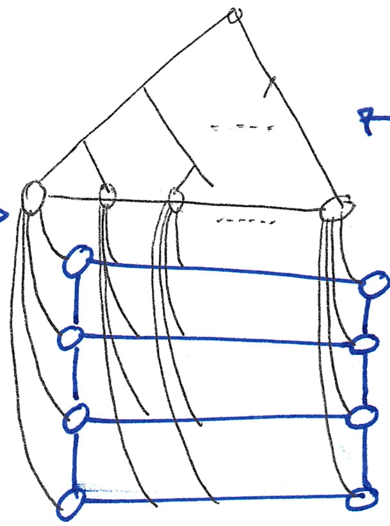
- REDUCE DIAMETER.

MODIFIED $G_{x,y}$:

LEVEL 0

d

ADD STARS CONNECTING ALL NODES IN EACH COLUMN



ADD TREE TO CONNECT STARS

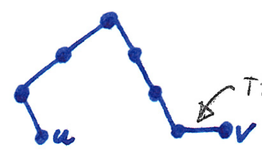
DIAM = $\Theta(\log n)$

AS TREE HAS $d = \log\left(\frac{n}{\lambda+1}\right)$ LEVEL

$\rightsquigarrow d \approx \frac{\lg n}{2}$ AS $\lambda = \sqrt{n}\lg n - 1$

DEF: • AN i-SHORTCUT BETWEEN NODES u,v IN LEVEL d IS A u,v-PATH OF LENGTH $2(d-i)\underline{+1}$. E.G.

ADDED FOR SIMPLICITY



IS A $[d-3]$-SHORTCUT

• WE SAY AN i-SC COVERS h HOPS IF h IS THE MAXIMAL DISTANCE BETWEEN u,v IN THE PATH AT LEVEL d.

CLAIM: • AN i-SC COVERS $\sqrt{n}/2^i\lg n$ HOPS AT LEVEL d.

• ALL i-SC COVER $\sqrt{n}/2$ HOPS AT LEVEL d.

PROOF:
$$\sum_{i=0}^{d} 2^i \cdot \sqrt{n}/2^i \lg n \geq \frac{d\sqrt{n}}{\lg n} \approx \frac{\sqrt{n}}{2} \qquad \square$$

LEM: ASSUME NODES CAN ONLY FORWARD RECEIVED MESSAGES. G_x AND G_y CANNOT EXCHANGE $\sqrt{n}\lg n$ BIT IN TIME $\leq \sqrt{n}/3\lg n$.

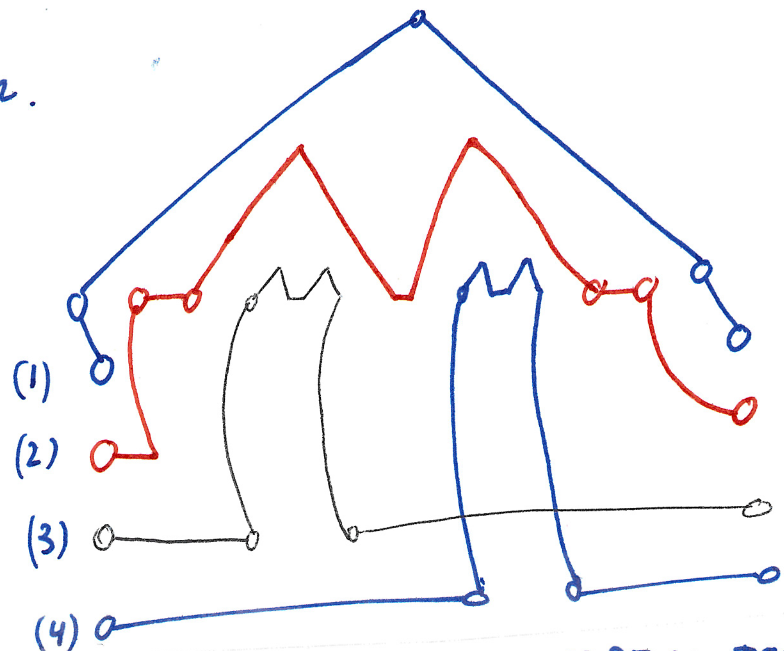
PROOF: # PATHS OF LENGTH $(\sqrt{n}/3\lg n)-1$ THAT THE SHORTCUTS CAN CONTRIBUTE:

$$\le \frac{\sqrt{n}/2}{\left(\underbrace{\sqrt{n}/\lg n}_{\text{NORMAL PATH-LENGTH } (n/\ell+1)} - \underbrace{\sqrt{n}/3\lg n}_{\text{DESIRED PATH-LENGTH}} + 1 - \text{"\# SC-EDGES USED"}\right)}$$

$\sqrt{n}/2$: # OF EDGES IN NORMAL PATHS THAT CAN BE COVERED.

$< 3\lg n.$

EXAMPLE:



⇝ CASES LIKE 3+4 CANNOT HAPPEN TOO OFTEN AT THE SAME TIME!

⇒ WITHIN TIME $\sqrt{n}/3\lg n - 1$ AT MOST

$$(\sqrt{n}/3\lg n - 1)\cdot 3\lg n \lneq \sqrt{n}$$

MESSAGES CAN BE EXCHANGED.

EACH MESSAGE CONTAINS $\lg n$ BIT.

(IF WE ASSUME $\Theta(\lg n)$ BIT WE NEED TO CHOOSE CONSTANTS DIFFERENTLY). □

NOTE: IN THE ABOVE ARGUMENT WE EVEN USE SEVERAL TREE-EDGES SEVERAL TIMES IN THE SAME TIMESLOT.

QUESTION: HOW DO WE REMOVE THE ASSUMPTION $x_i + \bar{y}_i \neq 0$?

DEF: $x' := x \circ \bar{x} = (x_1, \dots, x_k, \bar{x}_1, \dots, \bar{x}_k) \in \{0,1\}^{2k}$ FOR GIVEN x

$y' := \bar{y} \circ y = (\bar{y}_1, \dots, \bar{y}_k, y_1, \dots, y_k) \in \{0,1\}^{2k}$ " " y

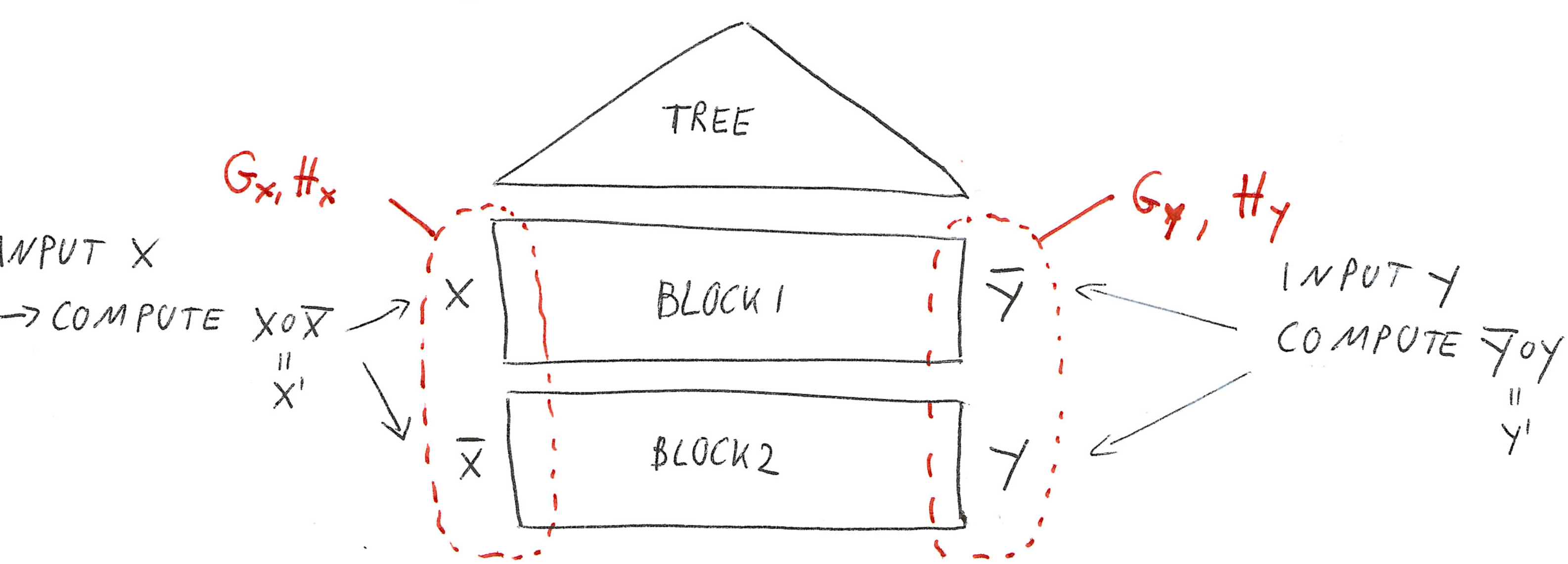
LEM: IFF $x \neq y$ THERE IS AN $j \in \{1, \dots, 2k\}$ s.t. $x'_j = y'_j = 0$

PROOF: CASE $x \neq y \Rightarrow \exists i \in \{1, \dots, k\}$ s.t. $x_i \neq y_i$

$\Rightarrow$ EITHER $\underbrace{x_i}_{= x'_i} = \underbrace{\bar{y}_i}_{= y'_i} = 0$ OR $\underbrace{\bar{x}_i}_{= x'_{i+k}} = \underbrace{y_i}_{= y'_{i+k}} = 0$

CASE $x = y \Rightarrow$ ALWAYS $x_i + \bar{y}_i = 1 = \bar{x}_i + y_i$ $\boxtimes$

FINAL CONSTRUCTION:



EXTEND H_{xy} s.t. A SPANNING TREE OF THE TREE IS INCLUDED AND THE BLOCKS 1&2 ARE CONNECTED BY ONE EDGE.

CLAIM: H_{xy} IS DISCONNECTED IFF $\exists i$ s.t. $x_i' = y_i' = 0$
IS TREE OTHERWISE.

THIS COMPETES THE PROOF OF LEM. II.

DEF: SPANNING TREE T $\alpha(n)$-APPROXIMATES MST T_{MST} IF

$$\omega(T_{MST}) \leq \omega(T) \leq \alpha(n) \cdot \omega(T_{MST})$$

THM: $\alpha(n)$-APPROXIMATING AN MST TAKES $\Omega(n+D)$ ROUNDS
└ e.g. poly(n)

PROOF: ASSIGN WEIGHT 1 (OR 0) TO EDGES IN H.
" " $\alpha(n) \cdot n$ " " NOT IN H.

- IF H IS TREE, THEN H IS MST
  $\Rightarrow$ MST HAS WEIGHT $n-1$
- IF H IS NOT A TREE, THEN H IS DISCONNECTED (BY CONSTRUCTION OF H)
  $\Rightarrow$ MST HAS WEIGHT $\geq \alpha(n) n + n - 2$

$\Rightarrow$ IF AN ALGORITHM COULD $\alpha(n)$-APPROXIMATE AN MST IT COULD DECIDE CORRECTLY IF H_{xy} IS A SPANNING TREE

THM: $\alpha(n)$-APPROXIMATING SSSP TAKES TIME $\tilde{\Omega}(\sqrt{n})$.

PROOF: USE THE ABOVE WEIGHTS.

- IF H IS A TREE, THEN ALL SHORTEST PATHS HAVE LENGTH $\leq n-1$.
- IF H IS NOT A TREE, THERE IS AT LEAST ONE SHORTEST PATH OF LENGTH $\alpha(n) \cdot n$. THIS CAN BE DETECTED IN TIME $O(D)$. □

NOTE: IN THE PROOF WE ASSUMED THAT NODES [NOT IN G_x, G_y] CAN ONLY FORWARD MESSAGES. USING SIMULATION ARGUMENTS THIS ASSUMPTION CAN BE REMOVED IN OUR CONSTRUCTION.

NOTE: USING THE SET-DISJOINTNESS FUNCTION INSTEAD OF EQ, ONE CAN EXTEND THESE DETERMINISTIC LOWER BOUNDS TO RANDOMIZED LOWER BOUNDS.

NOTE: IT TURNED OUT THAT IN THE CONGEST MODEL VERIFYING CAN BE HARDER THAN COMPUTING!

E.G. SPANNING TREE

- VERIFYING THAT A PROPOSED SOLUTION IS INDEED A SPANNING TREE TAKES $\tilde{\Omega}(\sqrt{n} + D)$
- COMPUTING A SOLUTION (BFS-TREE) TAKES $O(D)$.

COMPARE THIS TO CLASSIC COMPUTING THEORY:

E.G. SAT (SATISFYABILITY)

- VERIFYING AN ASSIGNMENT FOR SAT IS IN P
- COMPUTING A SOLUTION FOR SAT IS IN NP
- COMPUTING MIGHT BE HARDER THAN VERIFYING!