236358
Distributed Graph Algorithms

Spring 2017
Class by Keren Censor-Hillel
MST in CONGEST

- **BFS-based** algorithm in $O(n)$ rounds
- **GHS algorithm** in $O(n \log n)$ rounds
- **GKP algorithm** in $O(\sqrt{n} \log^* n + D)$ rounds

**Question:** $D$ is necessary. What about $\sqrt{n}$?
- We will see that $\sqrt{n}$ is also a lower bound
- But first, a simpler lower bound proof
Computing the Diameter

• A BFS tree gives a 2-approximation to the diameter
  \[ D = \max \{ d(u, v) \mid u, v \text{ in } V \} \]

• Can be computed in \( O(D) \) rounds in CONGEST

• What about exact diameter?
• Better approximation factors?
Diameter in CONGEST

• Exact diameter can be computed in $O(n)$ rounds
  – Even APSP (all-pairs-shortest-paths)

• Any algorithm for computing the exact diameter requires $\Omega(n/\log n)$ rounds
  – Even when $D$ is small
Approximating the Diameter in CONGEST

• A \((3/2-\epsilon)\)-approximation of \(D\) requires \(\Omega(n/\log^3 n)\) rounds

• A \(3/2\)-approximation of \(D\) can be computed in \(O((n/\log n)^{1/2}+D)\) rounds

• A threshold at \(3/2\)
Diameter in CONGEST

- **Theorem**: Any algorithm for computing the exact diameter requires $\Omega(n/\log n)$ rounds.

- **Proof**: Reduction from 2-party Set Disjointness.

- **Reminder**: If we reduce solving A to solving B, and we show that solving A is hard, then solving B is hard.
2-party communication

- Two players, Alice and Bob

- Inputs: $x^A = (x^{A_1}, \ldots, x^{A_k})$, $x^B = (x^{B_1}, \ldots, x^{B_k})$ in $\{0,1\}^k$

- Players exchange bits according to a protocol $\pi$

- Outputs: $y^A, y^B$ in $\{0,1\}$
2-party communication

• At the beginning and after each bit that is sent in the protocol $\pi$:
  – Both players know the bit
  – Both players know whether the protocol is finished (and their output) or who sends the next bit

• The sequence of sent bits in $\pi$ is the transcript
2-party communication

• The *communication complexity* of a protocol $CC(\pi)$: Total number of bits sent in the transcript

• The *communication complexity* of a problem: The minimal complexity over all protocols that solve the problem
2-party communication

• **Set Disjointness:**

• **Inputs:**
\[ x^A = (x^A_1, \ldots, x^A_k), \quad x^B = (x^B_1, \ldots, x^B_k) \text{ in } \{0,1\}^k \]

• **Outputs:**
\[ y^A = y^B = 1 \text{ if and only if there is an } i \in \{1, \ldots, k\} \text{ such that } x^A_i = x^B_i = 1 \]
Lower Bound for Set Disjointness

- **Theorem**: The communication complexity of set disjointness is $k+1$

- **Proof**:
  - There is a protocol that solves set disjointness with $CC(\pi)=k+1$
    - Alice sends all of her input bits to Bob ($k$ bits)
    - Bob sends Alice the result (1 bit)
Lower Bound for Set Disjointness

- **Theorem**: The communication complexity of set disjointness is $k+1$

- **Proof**:  
  - Any protocol $\pi$ that solves set disjointness has $\text{CC}(\pi)=k+1$
    - Later if we have time
Computing D – Base Graph $G_{base}$
Computing D

Clique A\(^1\)

Clique B\(^1\)

Clique A\(^2\)

Clique B\(^2\)

a

b
Computing D

Clique $A^1$          Clique $B^1$

Clique $A^2$          Clique $B^2$

a                      b
Computing $D$

Clique $A^1$  Clique $B^1$

Clique $A^2$  Clique $B^2$
Input-Based Graph

• Given an input $x^A=(x^A_1,\ldots, x^A_k)$, $x^B=(x^B_1,\ldots, x^B_k)$ to Set Disjointness, define $G$ as follows:
  
  – $G$ contains all of $G_{\text{base}}$
  
  – For $(i,j)=1,\ldots,k$, the spike from $A^1_i$ to $A^2_j$ is in $G$ if and only if $x^A_{i,j}=0$
    
    • $(i,j)=(i-1)\sqrt{k}+j$
  
  – For $(i,j)=1,\ldots,k$, the spike from $B^1_i$ to $B^2_j$ is in $G$ if and only if $x^B_{i,j}=0$
Computing D – Input-Based Graph

Clique A<sup>1</sup>  Clique B<sup>1</sup>

Clique A<sup>2</sup>  Clique B<sup>2</sup>
Computing D – Input-Based Graph
Computing D – Input-Based Graph

\[ \text{Clique A}^1 \quad \text{Clique B}^1 \]

\[ \text{Clique A}^2 \quad \text{Clique B}^2 \]
Computing D

• **Claim 1**: If the inputs are disjoint then $D(G)=2$. Otherwise, $D(G)=3$.

• **Proof**: By case analysis.
Diameter in CONGEST

• **Theorem**: Any algorithm for computing the exact diameter requires $\Omega(n/\log n)$ rounds.


• **Illustration**
Computing D

Clique A\(^1\)

Clique B\(^1\)

Clique A\(^2\)

Clique B\(^2\)

a

b
Diameter in CONGEST

• **Theorem**: Any algorithm for computing the exact diameter requires $\Omega(n/\log n)$ rounds

• **Proof** (cont.):
  At most $O(n\log n)$ bits can be sent over the cut in a round. But $k=\Theta(n^2)$, so the number of rounds is $\Omega(n^2/n\log n)=\Omega(n/\log n)$. 

Keren Censor-Hillel, Spring 2017
(3/2-\(\varepsilon\))-Approximation of D

• **Theorem**: Any algorithm for computing a (3/2-\(\varepsilon\))-approximation of the diameter requires \(\Omega(n^{1/2}/\log n)\) rounds

• **Illustration**
(3/2-\(\epsilon\))-Approximation of D

Clique A

Clique A\(^1\)

Clique A\(^2\)

Clique B

Clique B\(^1\)

Clique B\(^2\)

a

b
(3/2-\(\varepsilon\))-Approximation of D

- **Theorem**: Any algorithm for computing a (3/2-\(\varepsilon\))-approximation of the diameter requires \(\Omega(n^{1/2}/\log n)\) rounds.

- **Proof**: Reduction from Set Disjointness. Alice simulates \(a\) and nodes in \(A^1, A^2\) and \(A^3\), Bob simulates \(b\) and nodes in \(B^1, B^2\) and \(B^3\).
(3/2-ε)-Approximation of D

• **Theorem**: Any algorithm for computing a (3/2-ε)-approximation of the diameter requires $\Omega(n^{1/2}/\log n)$ rounds.

• **Proof**: At most $O(n^{1/2}\log n)$ bits can be sent over the cut in a round. But now $k=\Theta(n)$, so the number of rounds is $\Omega(n/n^{1/2}\log n) = \Omega(n^{1/2}/\log n)$. 
Lower Bound for Set Disjointness

• **Theorem**: The communication complexity of set disjointness is $k+1$

• **Proof**:
  • Any protocol $\pi$ that solves set disjointness has $\text{CC}(\pi)=k+1$
  • Will be proven by **Claims 1** and **2** next.
Rectangles

• Given two subsets $X,Y$ of $2^k$, the set $X \times Y$ is called a rectangle.

• Illustration

• Given a function $f:2^k \times 2^k \rightarrow \{0,1\}$, a rectangle $X \times Y$ is $f$-monochromatic if $f(x,y) = f(x',y')$ for all $x,x' \in X$ and $y,y' \in Y$. 
Monochromatic Rectangles

• **Claim 1**: A protocol $\pi$ with $\mathbb{CC}(\pi)=b$ divides the input domain $2^k \times 2^k$ to $2^b$ disjoint monochromatic rectangles

• **Proof**: by induction.

• **Base case $b=0$**: an empty protocol gives the same output for every set of inputs.
Monochromatic Rectangles

• **Induction hypothesis**: A protocol \( \pi \) with \( \text{CC}(\pi) = b-1 \) divides the input domain \( 2^k \times 2^k \) to \( 2^{b-1} \) disjoint monochromatic rectangles
Monochromatic Rectangles

• Induction step: Let $\pi$ be a protocol with $\text{CC}(\pi) = b$. Let $XxY$ be a rectangle that corresponds to a string $s$ of $b-1$ bits of the protocol.

• Assume that Bob sends the next bit.
Monochromatic Rectangles

• Assume that Bob sends the next bit.
  – $Y'$ in $Y$: the inputs for which the next bit is 0
  – $Y''$ in $Y$: the inputs for which the next bit is 1

• $XxY'$ and $XxY''$ are two monochromatic rectangles corresponding to $s$

• $2^{b-1}$ strings $s$ give $2^b$ monochromatic rectangles
Rectangles in Set Disjointness

• **Claim 2**: A protocol $\pi$ that solves Set Disjointness divides the input domain $2^k \times 2^k$ to at least $2^k+1$ disjoint monochromatic rectangles.

• **Proof**: Consider the $2^k$ pairs of the form $(u,\bar{u})$. The output for all of them is 0.
Rectangles in Set Disjointness

• **Claim 2**: A protocol $\pi$ that solves Set Disjointness divides the input domain $2^k \times 2^k$ to at least $2^k + 1$ **disjoint monochromatic rectangles**

• **Proof**: No two such pairs $(u, \bar{u})$ and $(u', \bar{u}')$ can be in the same monochromatic rectangle because then the output for $(u, \bar{u'})$ and $(u', \bar{u})$ is also 0, but it has to be 1 for one of them.
Rectangles in Set Disjointness

- **Claim 2**: A protocol $\pi$ that solves Set Disjointness divides the input domain $2^k \times 2^k$ to at least $2^k + 1$ disjoint monochromatic rectangles.

- **Proof**: This gives $2^k$ disjoint monochromatic rectangles.
- There is at least one more additional rectangle for outputs 1.
Lower Bound for Set Disjointness

• **Theorem**: The communication complexity of **set disjointness** is **k+1**

• **Proof**:
  • Any protocol \( \pi \) that solves set disjointness has \( \text{CC}(\pi) = k+1 \)
  • By **Claims 1** and **2**, \( \pi \) needs at least \( \log(2^k+1) \) bits, which is at least **k+1** bits.
Next Class

• The promised $\sim n^{1/2}$ lower bound for MST in CONGEST