graph

nodes
graph

nodes + edges
graph problems

vertex cover
graph problems

vertex cover
graph problems

vertex cover
graph problems

vertex cover
graph algorithms

input

output
graph algorithms

“distributed” vs. “centralised”
The centralised algorithm for the graph $G = (V, E)$:

- $V = \{1, 2, \ldots\}$
- $E = \{\{1,3\}, \ldots\}$

Cluster $C = \{3, 7, \ldots\}$ is computed using a centralised algorithm.
all input in one location

\[ G = (V, E) \]
\[ V = \{1, 2, \ldots\} \]
\[ E = \{\{1,3\}, \ldots\} \]

centralised algorithm

all output in one location

\[ C = \{3, 7, \ldots\} \]
all input in **one location**

time unit $\approx$ **one step of computation**

all output in **one location**

$G = (V,E)$

$V = \{1, 2, \ldots\}$

$E = \{\{1,3\}, \ldots\}$

$C = \{3, 7, \ldots\}$
distributed graph algorithms

graph = computer network
node = computer
edge = communication link
time = communication steps
graph: computer network

node: computer
initial information

\[ t = 0 \]
initial information

$t = 0$
time step: communication

\[ t = 1 \]
time step: communication
all nodes in parallel

t = 2
local outputs

$t = 2$

"0"

"0"

"1"
nodes that output “1”

vertex cover
distributed algorithm

map from radius-$t$ neighbourhoods to local outputs
distributed algorithm

trivial: \( t \geq \text{diameter} \)

focus: small \( t \)
our research: **local algorithms**, $t = O(1)$
“what can be computed locally?”
(Naor & Stockmeyer 1995)
“what can be computed locally?”

• **fast** and **fault-tolerant** distributed algorithms

• understanding social networks, markets, biological systems, ...
local algorithms

• **vertex cover:** 2-approx.
• edge dominating sets
• almost stable matchings
• linear programming...

(bounded-degree graphs)
local algorithms

matching lower bounds!

general proof techniques

e.g.: unique identifiers do not help with local approximation

decision problems...
distributed graph algorithms

local algorithms: $O(1)$ time

Thanks!