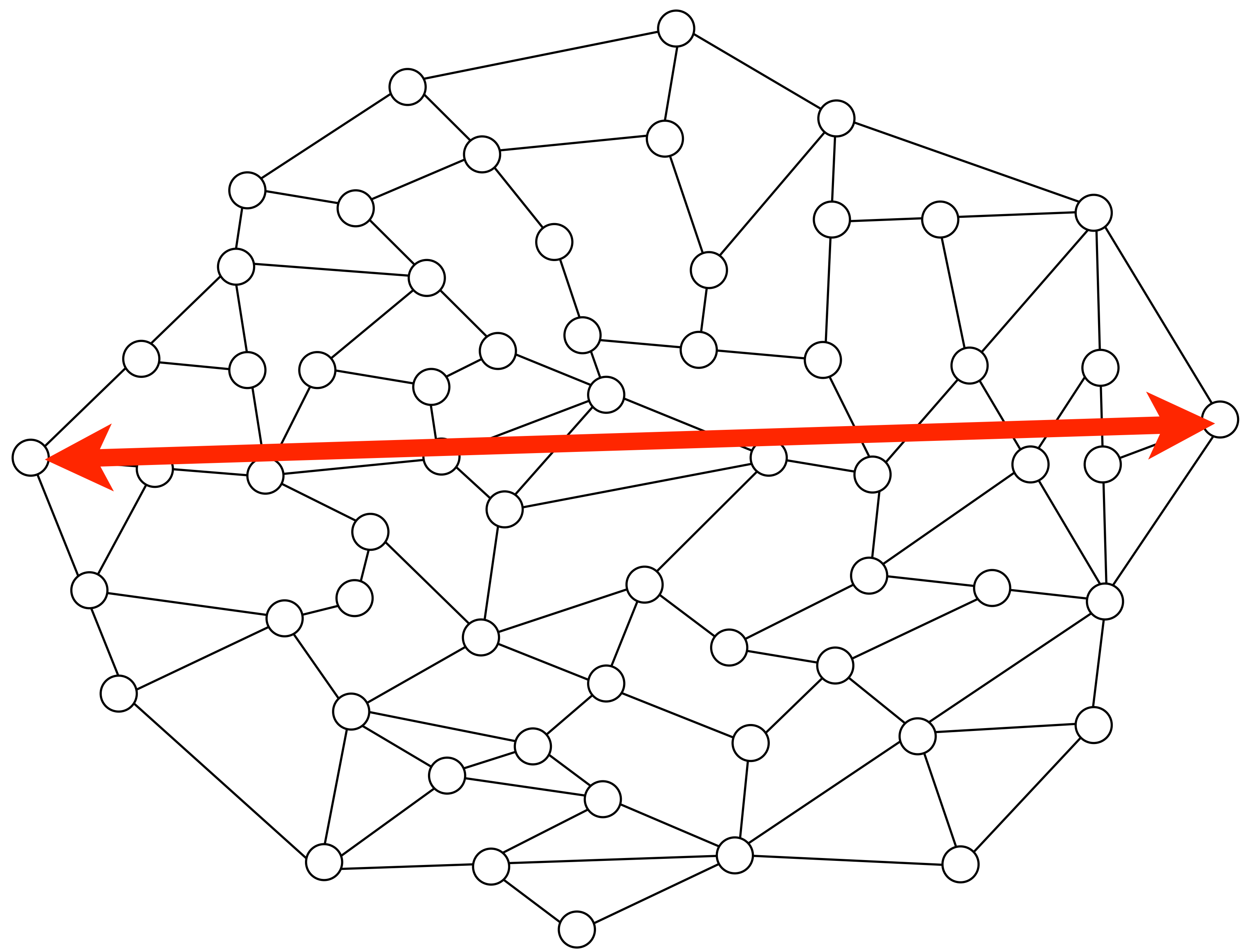
Approximating the Diameter of Planar Graphs in Near Linear Time

Oren Weimann

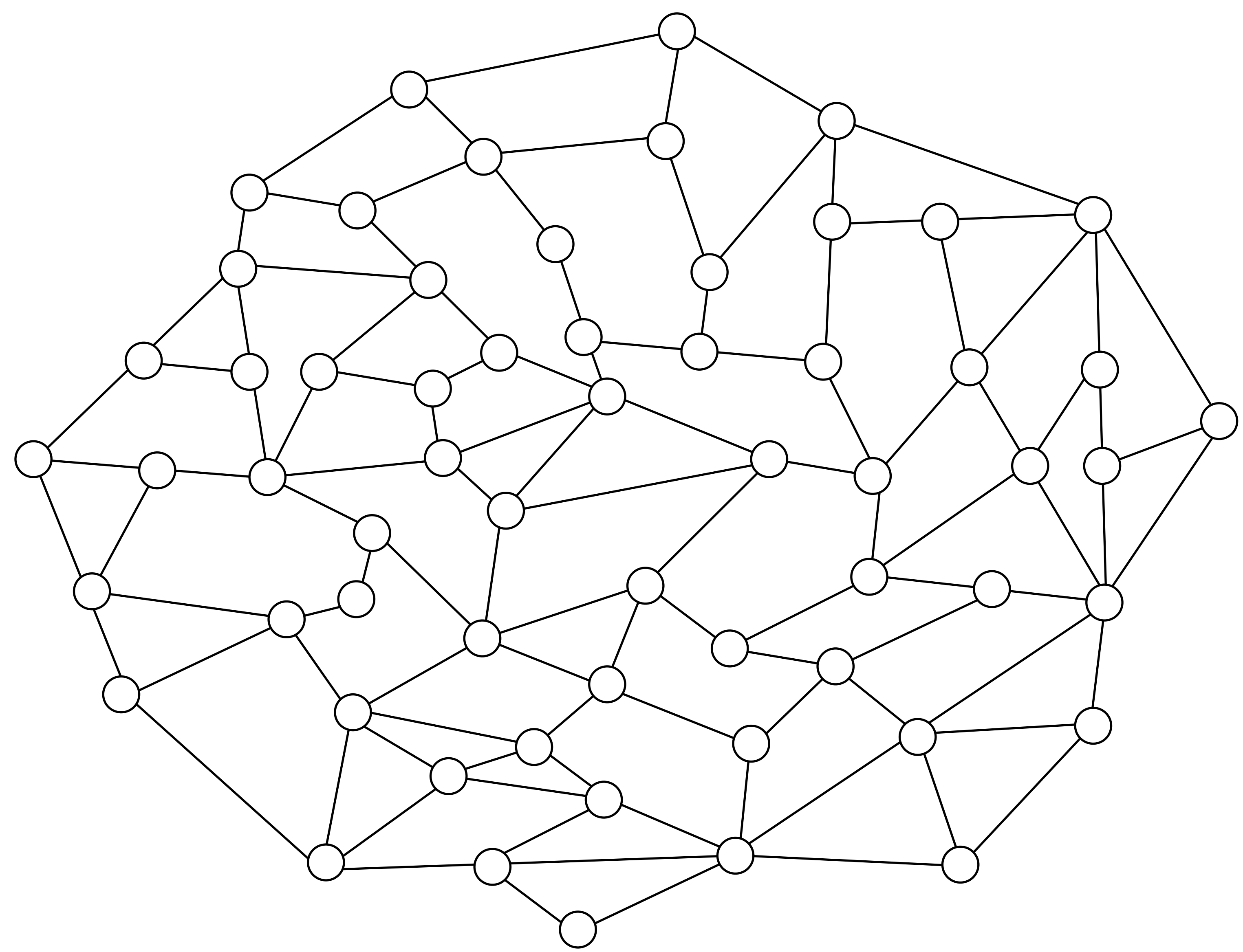
Raphael Yuster

אוניברסיטת חיפה
University of Haifa



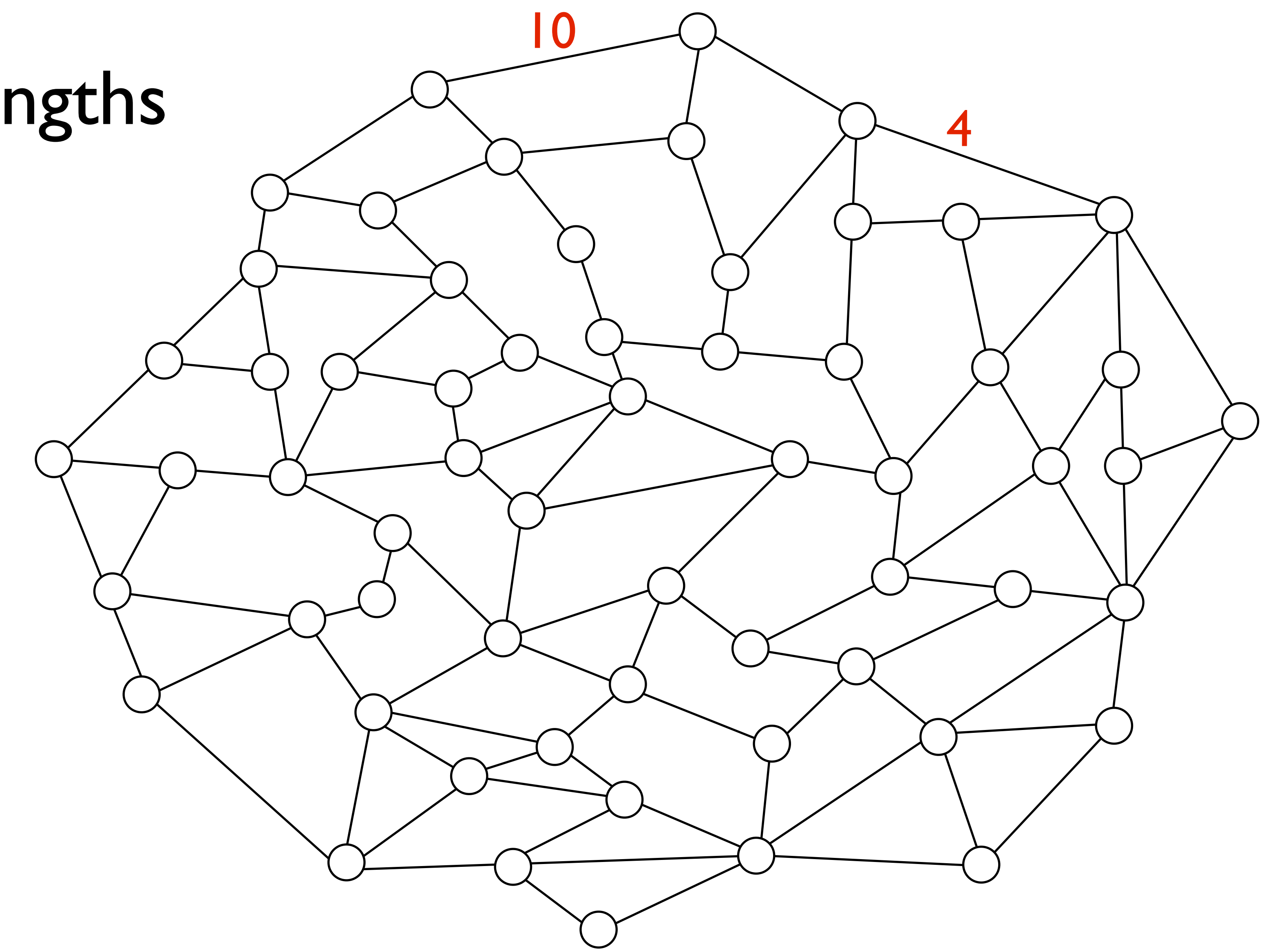
The Diameter Problem

- Planar graph
- Undirected



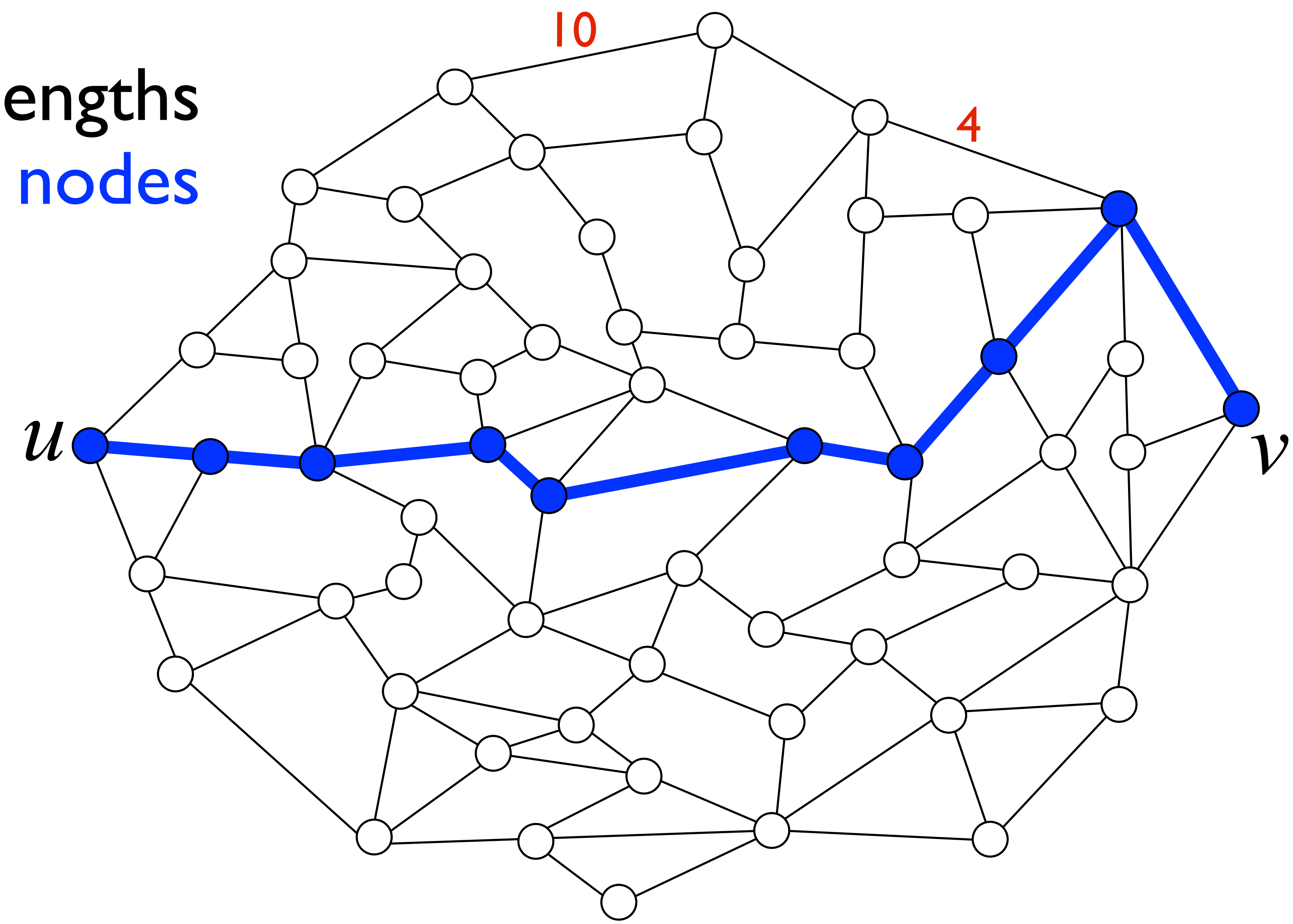
The Diameter Problem

- Planar graph
- Undirected
- Non-negative edge-lengths



The Diameter Problem

- Planar graph
- Undirected
- Non-negative edge-lengths
- Find furthest pair of nodes



Related Work

General graphs:

- APSP in $\tilde{O}(n^3)$ (faster for sparse graphs or small edge-lengths)
- Open: Diameter faster than APSP?

Related Work

General graphs:

- APSP in $\tilde{O}(n^3)$ (faster for sparse graphs or small edge-lengths)
- Open: Diameter faster than APSP?

Planar graphs:

- APSP in optimal $O(n^2)$ [Frederickson 1987]
- Diameter in $O(n^2 (\log\log n)^4 / \log n)$ [Wulff-Nilsen 2008]
- Open: Diameter in $O(n^{2-\varepsilon})$? [Chung 1987]
- Diameter in $O(n)$ for fixed diameter [Eppstein 1995]

Related Work

General graphs:

- APSP in $\tilde{O}(n^3)$ (faster for sparse graphs or small edge-lengths)
- Open: Diameter faster than APSP?

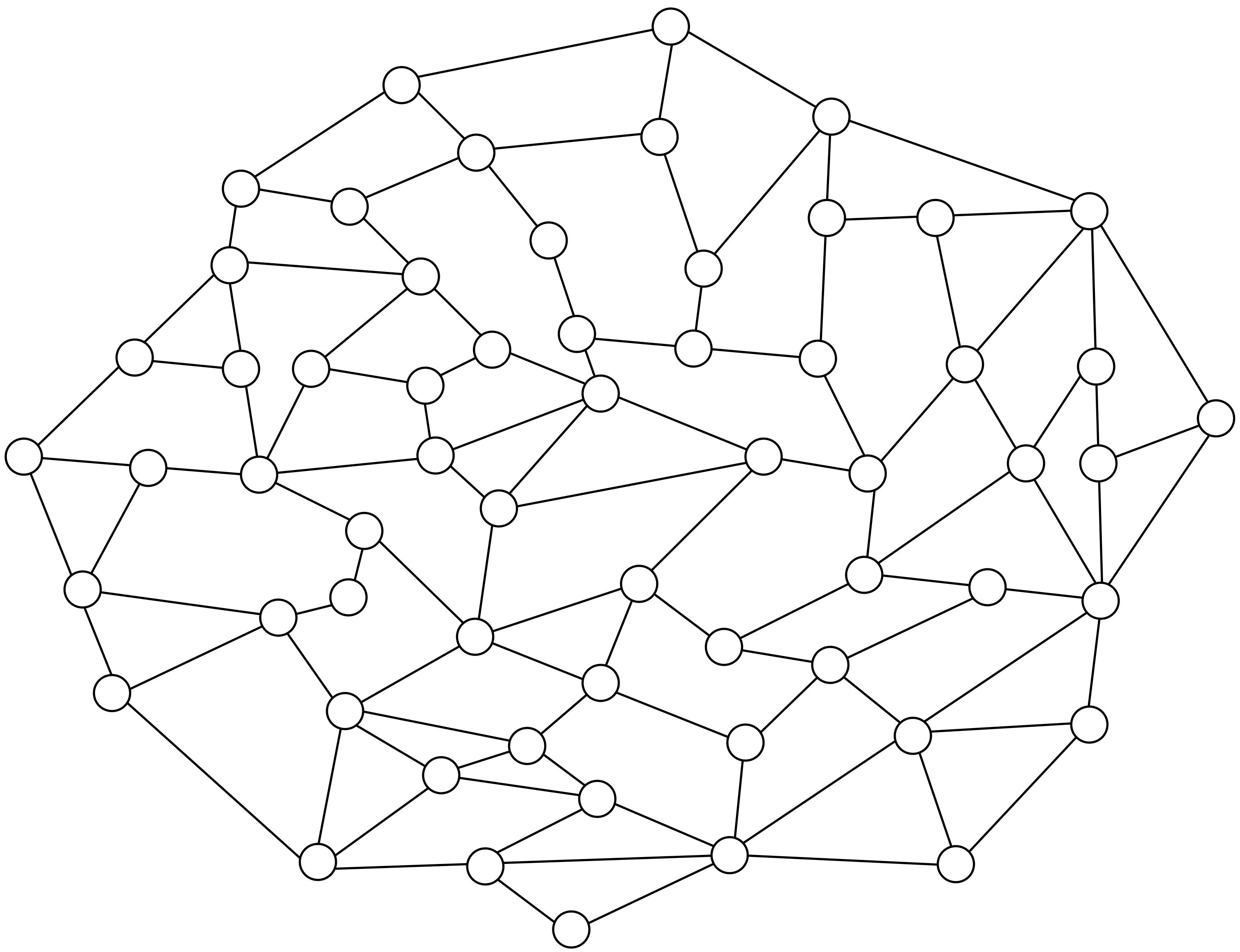
Planar graphs:

- APSP in optimal $O(n^2)$ [Frederickson 1987]
- Diameter in $O(n^2 (\log\log n)^4 / \log n)$ [Wulff-Nilsen 2008]
- Open: Diameter in $O(n^{2-\varepsilon})$? [Chung 1987]
- Diameter in $O(n)$ for fixed diameter [Eppstein 1995]

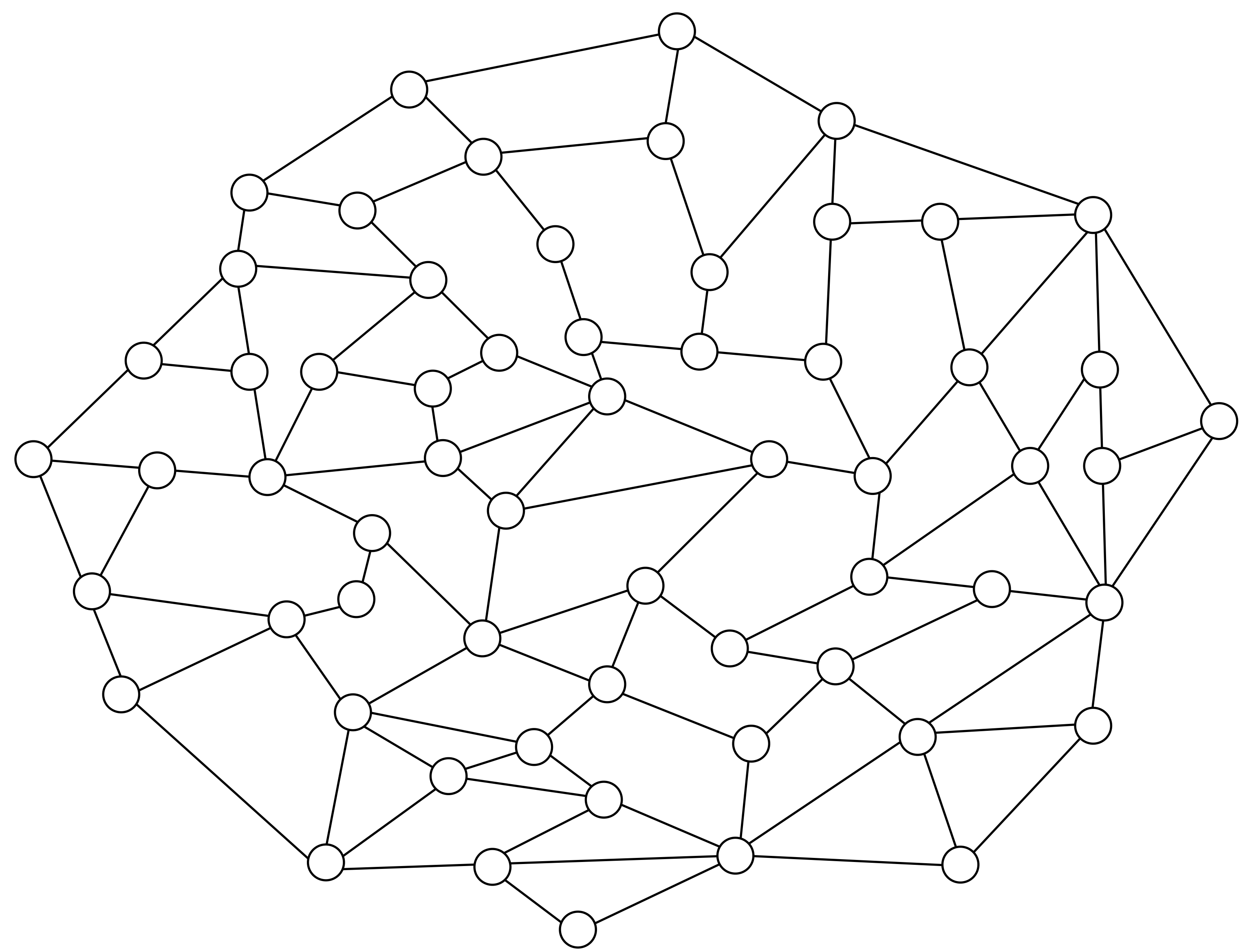
Planar graphs approximation:

- 2-approximation in $O(n)$ by SSSP tree [Henzinger et al. 1997]
- 1.5-approximation in $O(n^{1.5})$ [Berman et al. 2007]
- $(1+\varepsilon)$-approximation in $\tilde{O}(n)$ for any fixed $\varepsilon<1$

The Algorithm

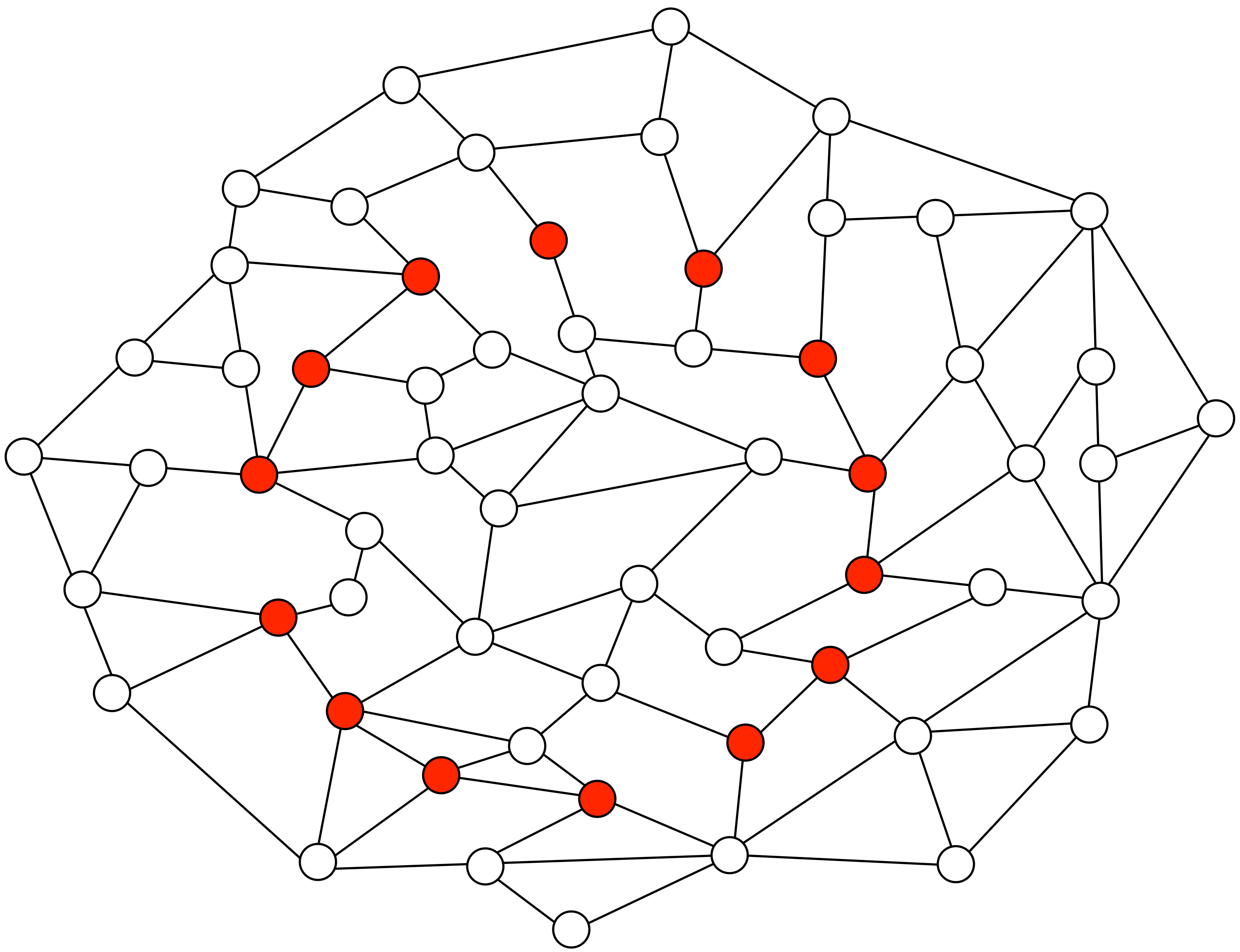


Planar Separator



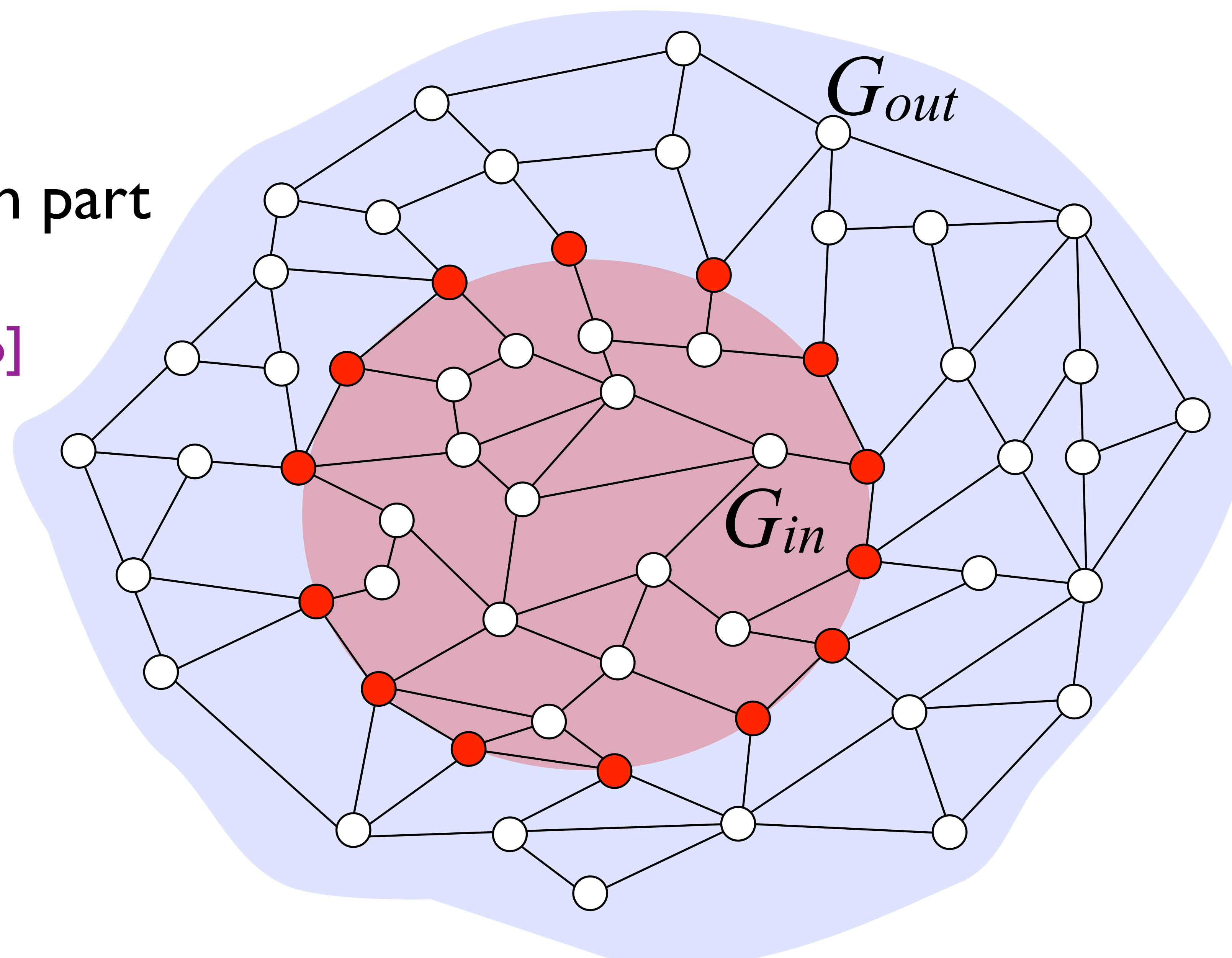
Planar Separator

- $O\left(\sqrt{n}\right)$ boundary nodes

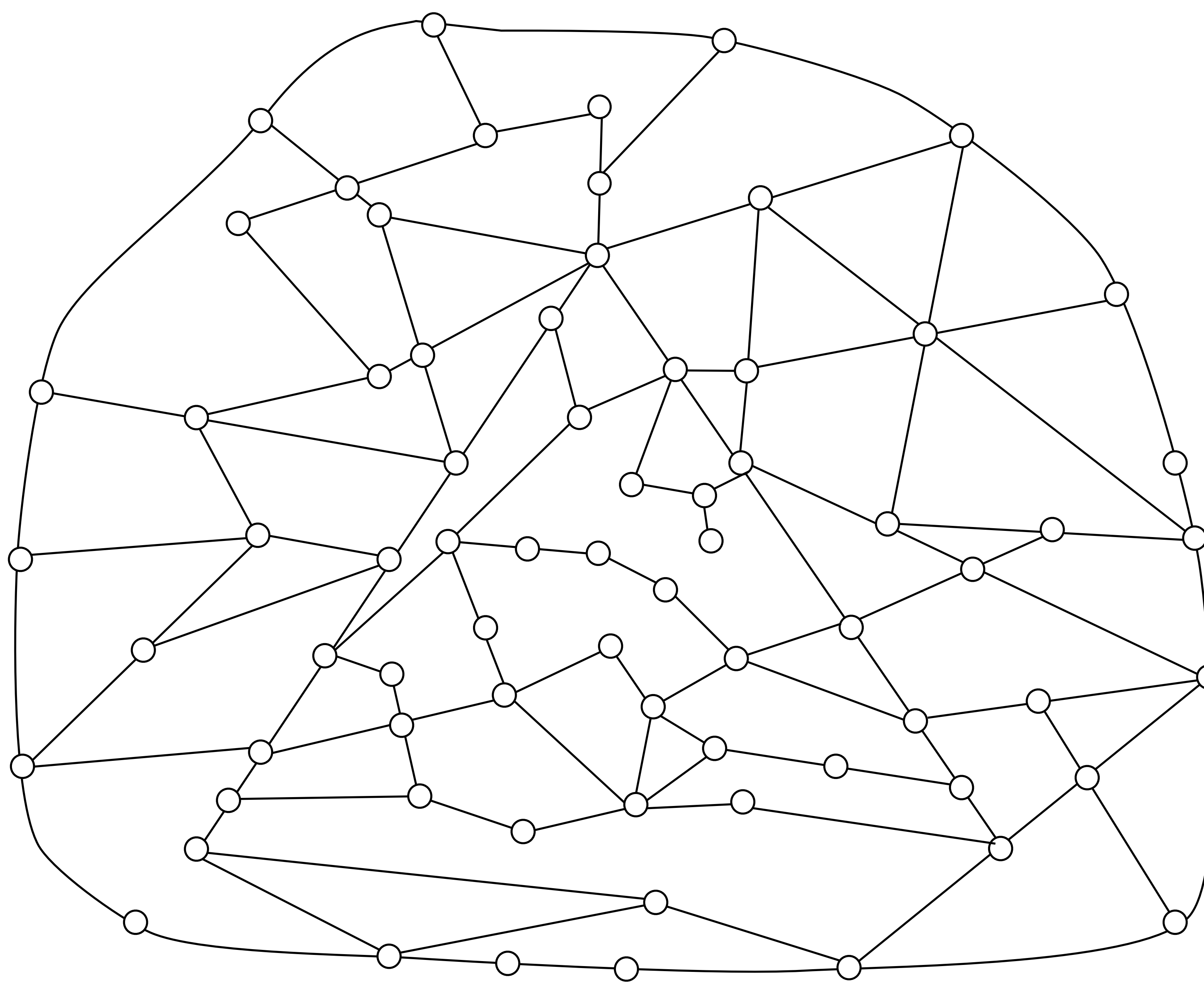


Planar Separator

- $O\left(\sqrt{n}\right)$ boundary nodes
- At most $2n/3$ nodes in each part
- Can be found in $O(n)$ time
  [Lipton-Tarjan 1979, Miller 1986]

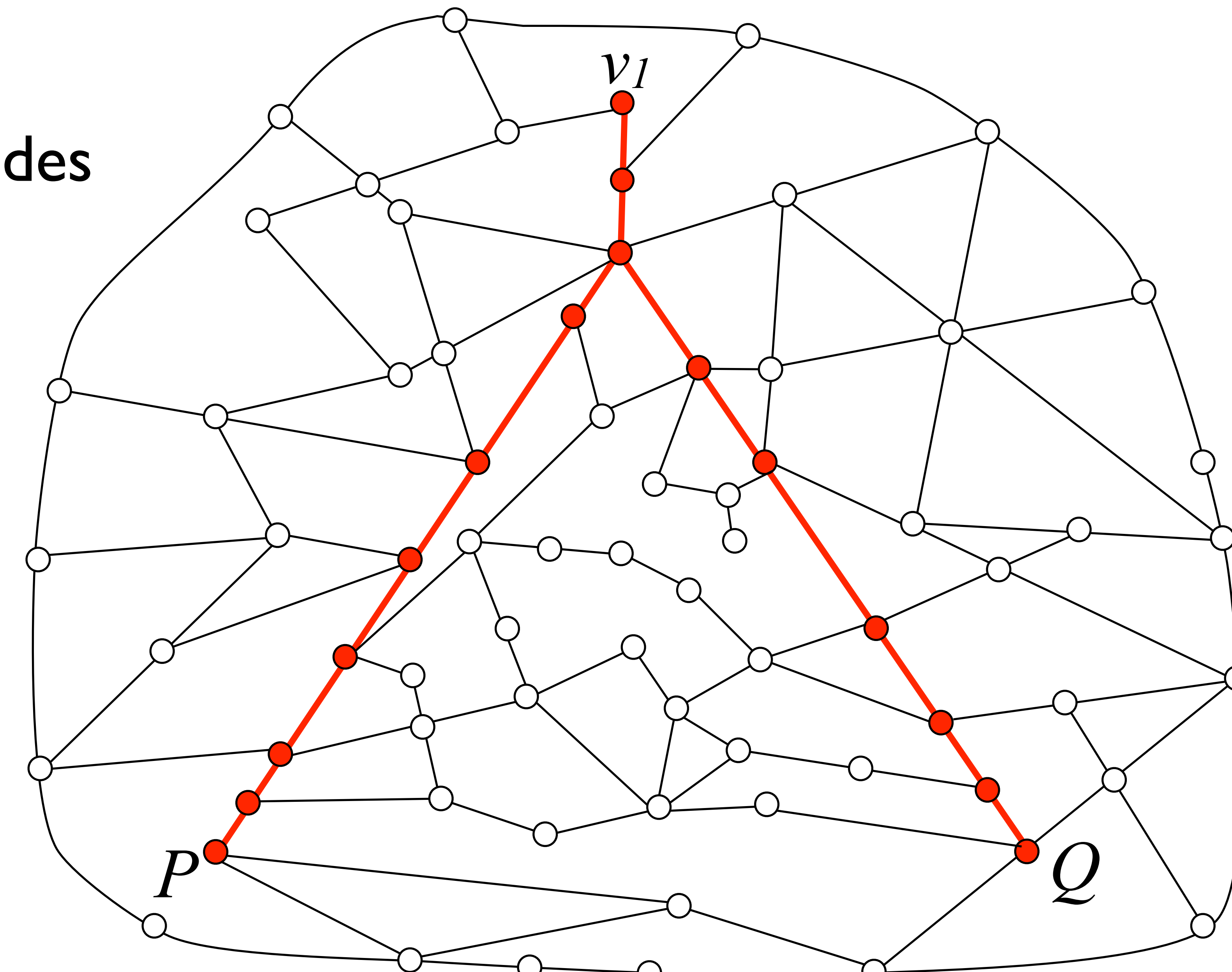


Planar <u>Shortest Path</u> Separator



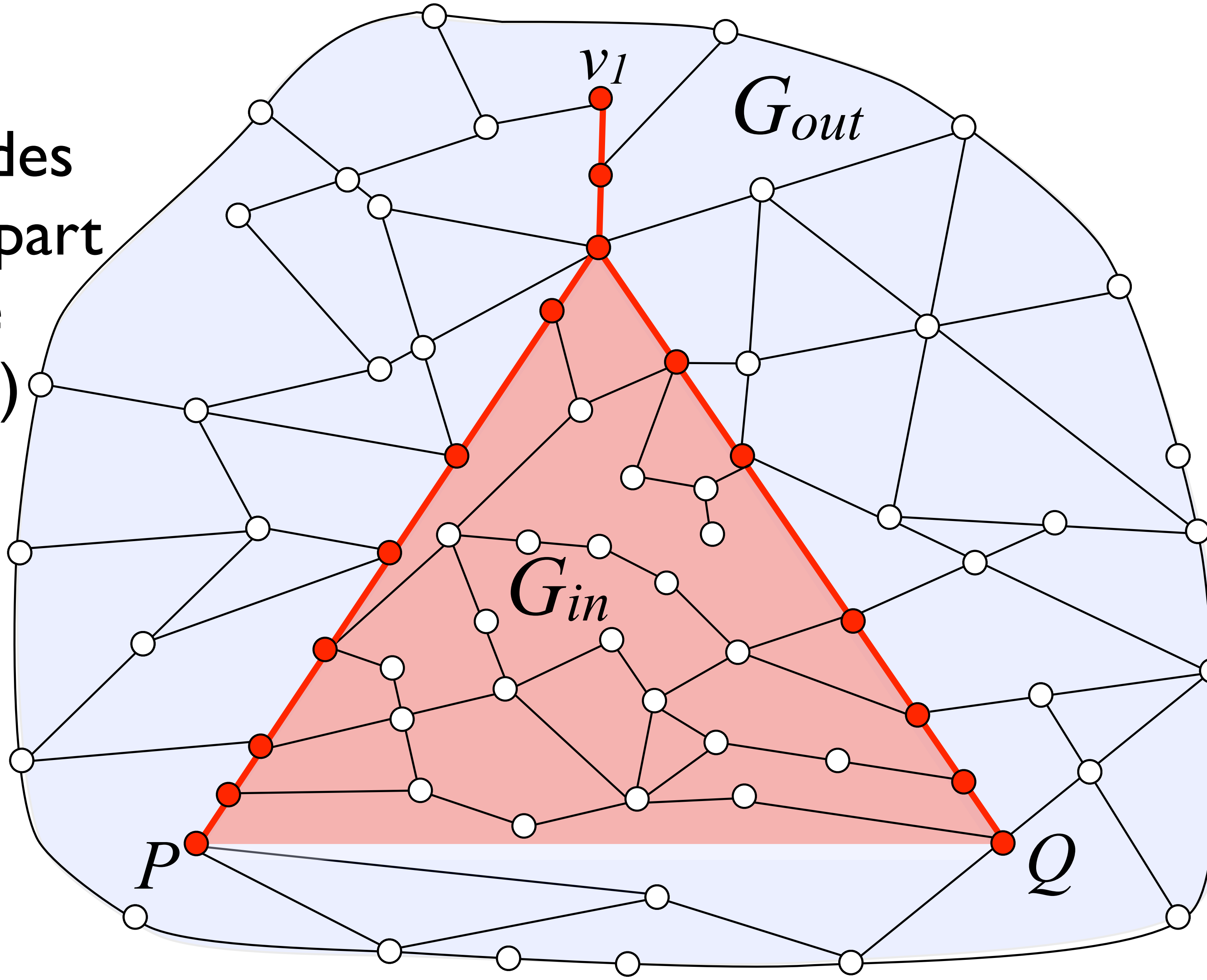
Planar Shortest Path Separator

- Can have $\Omega(n)$ boundary nodes



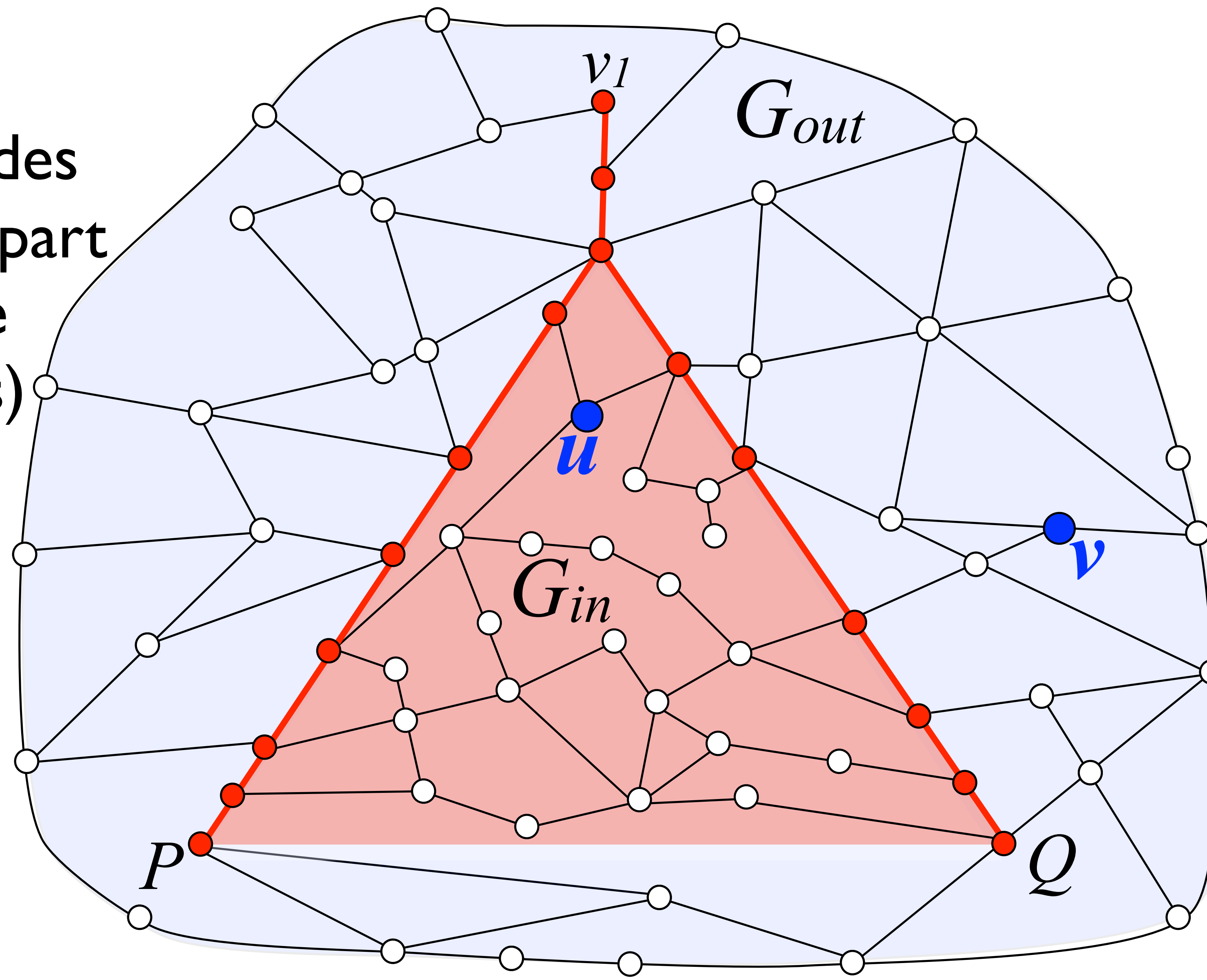
Planar <u>Shortest Path</u> Separator

- Can have $\Omega(n)$ boundary nodes
- At most $2n/3$ nodes in each part
- G_{in} and G_{out} both include the boundary (may have n nodes)



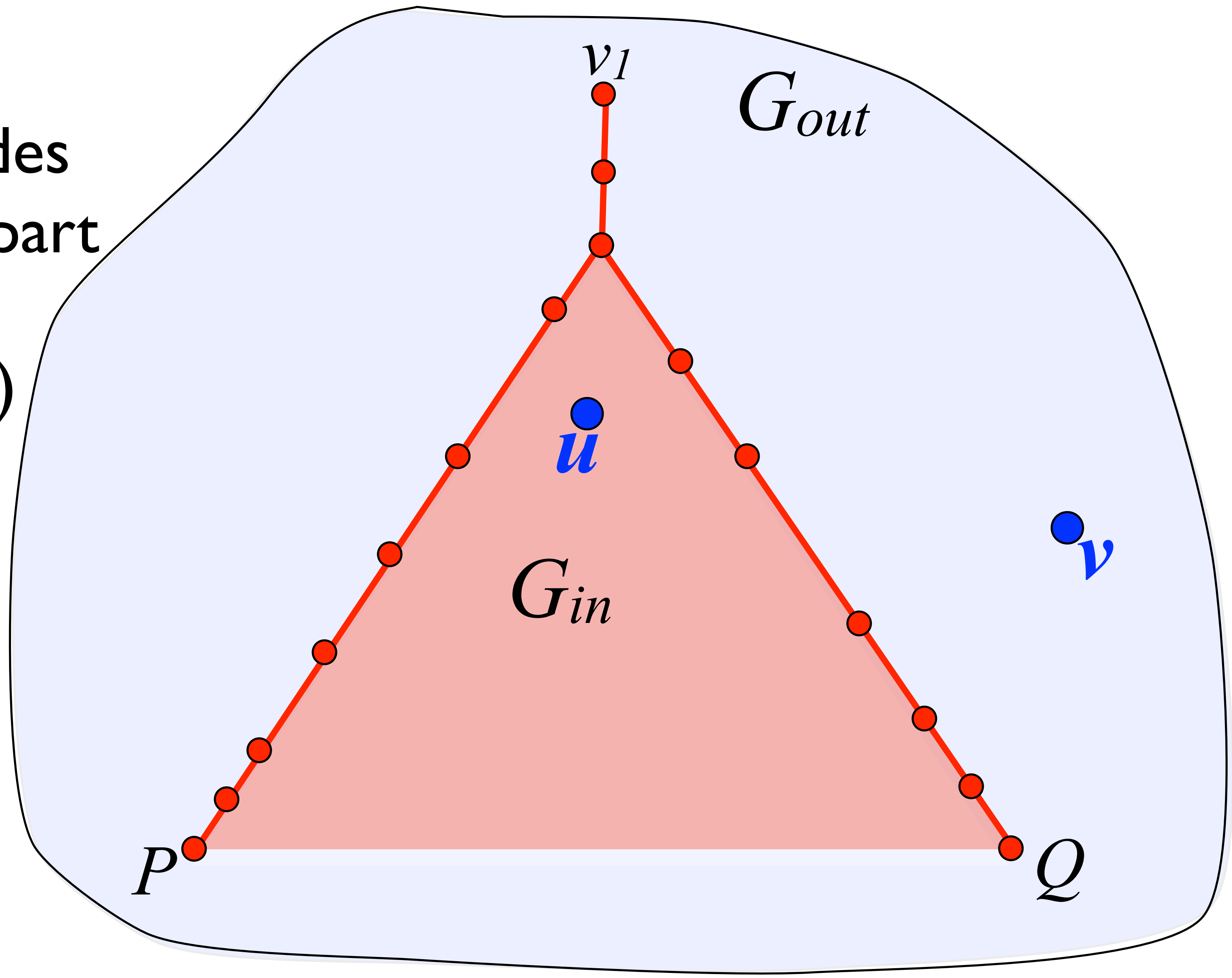
Planar Shortest Path Separator

- Can have $\Omega(n)$ boundary nodes
- At most $2n/3$ nodes in each part
- G_{in} and G_{out} both include the boundary (may have n nodes)



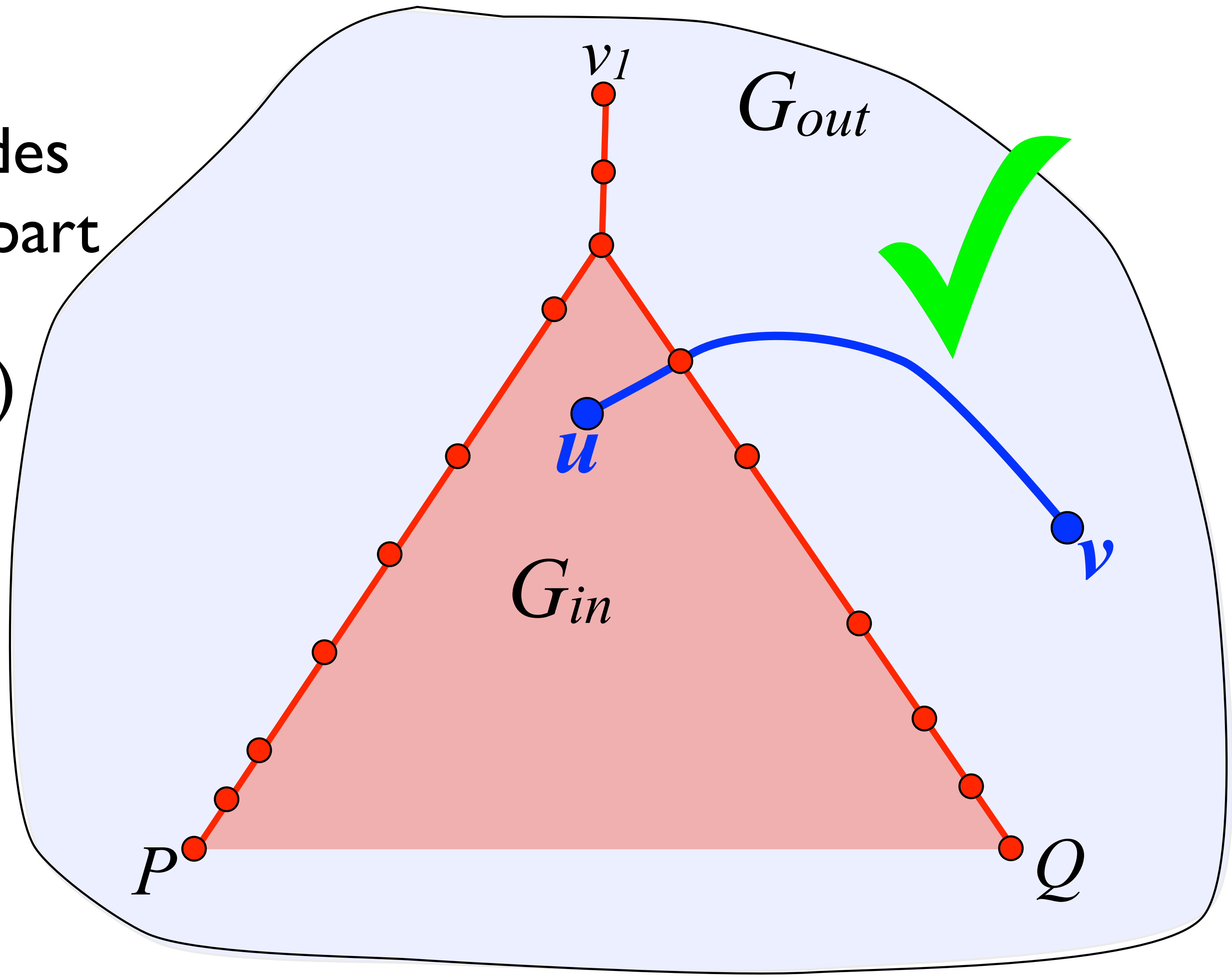
Planar Shortest Path Separator

- Can have $\Omega(n)$ boundary nodes
- At most $2n/3$ nodes in each part
- G_{in} and G_{out} both include the boundary (may have n nodes)



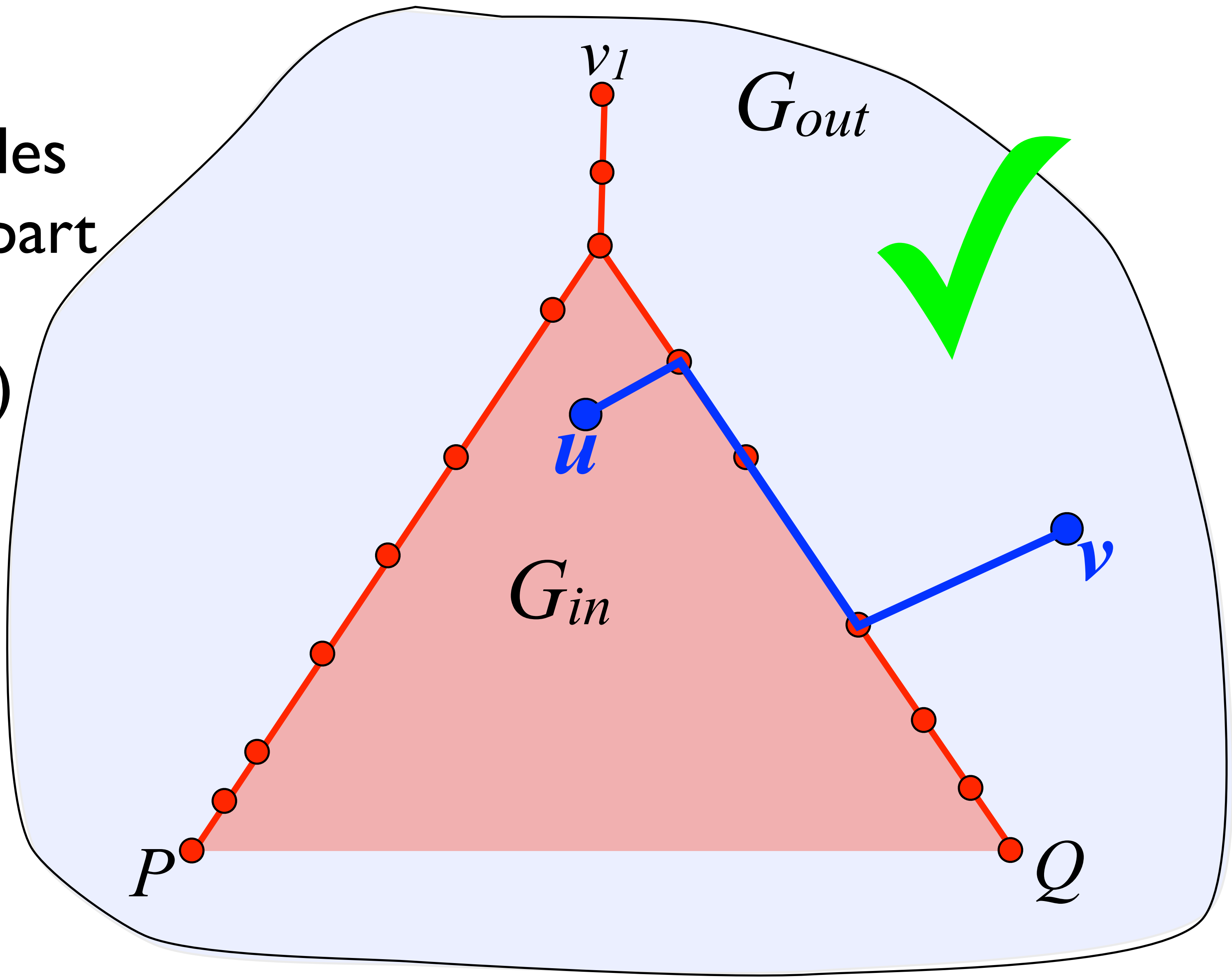
Planar Shortest Path Separator

- Can have $\Omega(n)$ boundary nodes
- At most $2n/3$ nodes in each part
- G_{in} and G_{out} both include the boundary (may have n nodes)



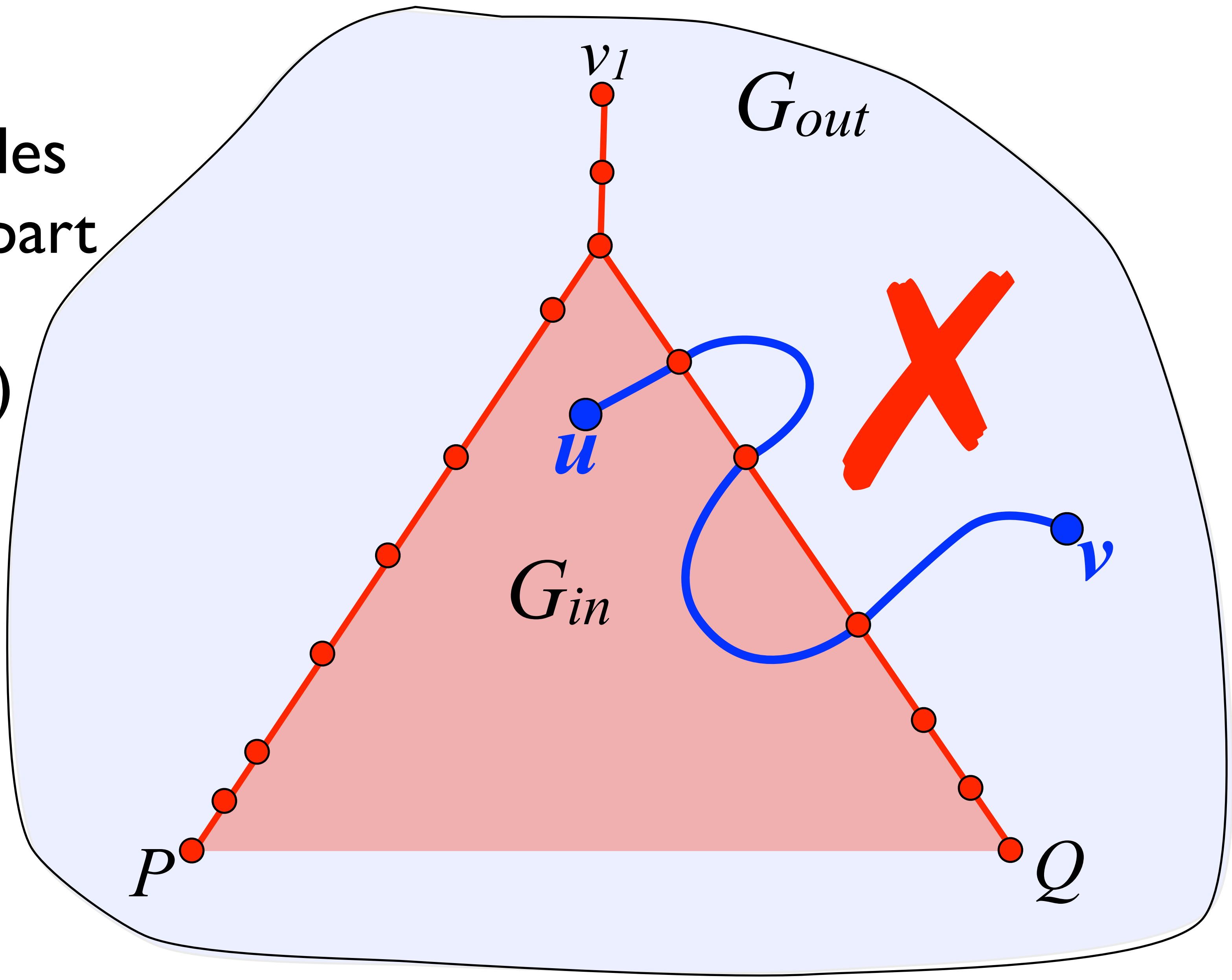
Planar Shortest Path Separator

- Can have $\Omega(n)$ boundary nodes
- At most $2n/3$ nodes in each part
- G_{in} and G_{out} both include the boundary (may have n nodes)

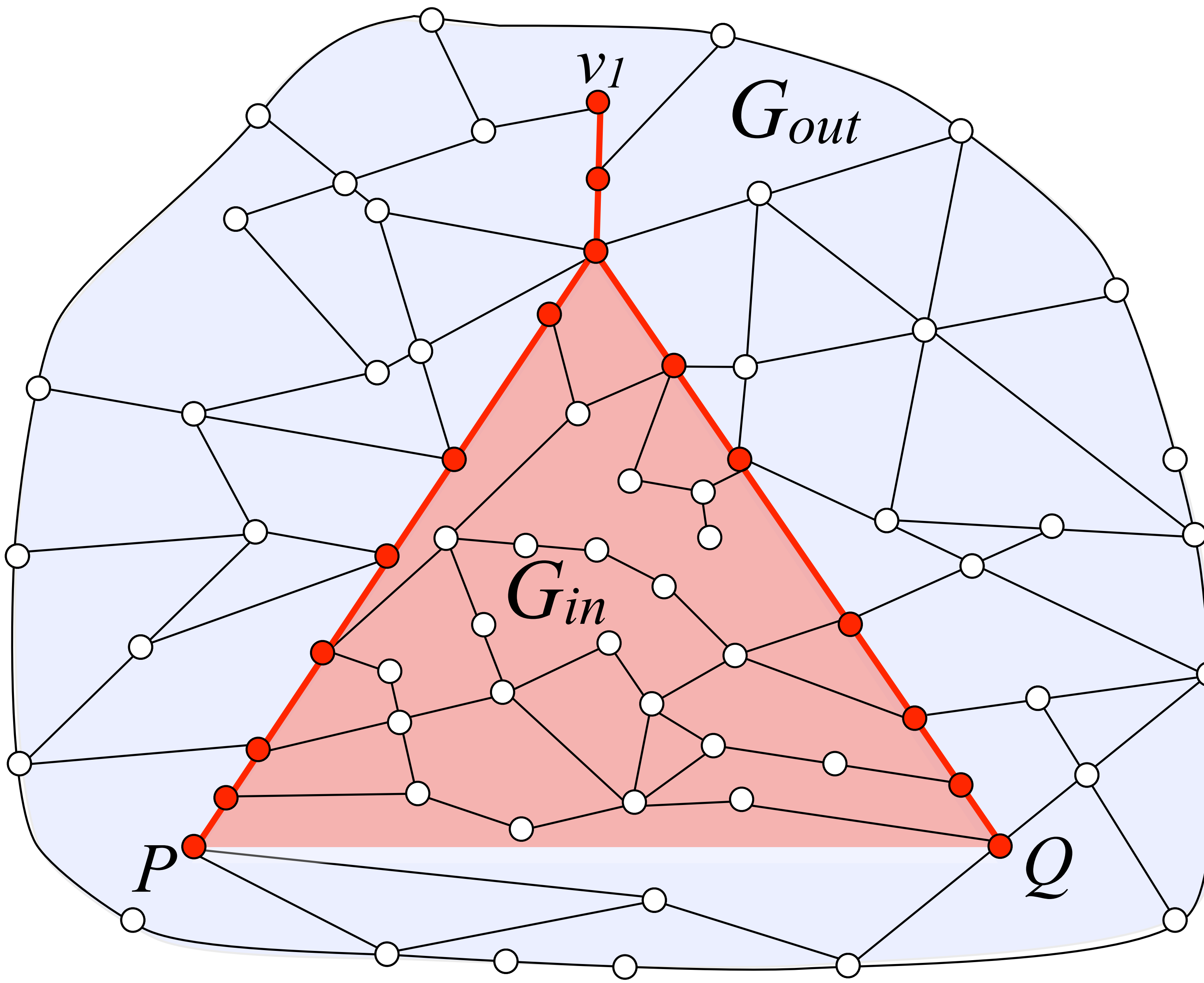


Planar Shortest Path Separator

- Can have $\Omega(n)$ boundary nodes
- At most $2n/3$ nodes in each part
- G_{in} and G_{out} both include the boundary (may have n nodes)

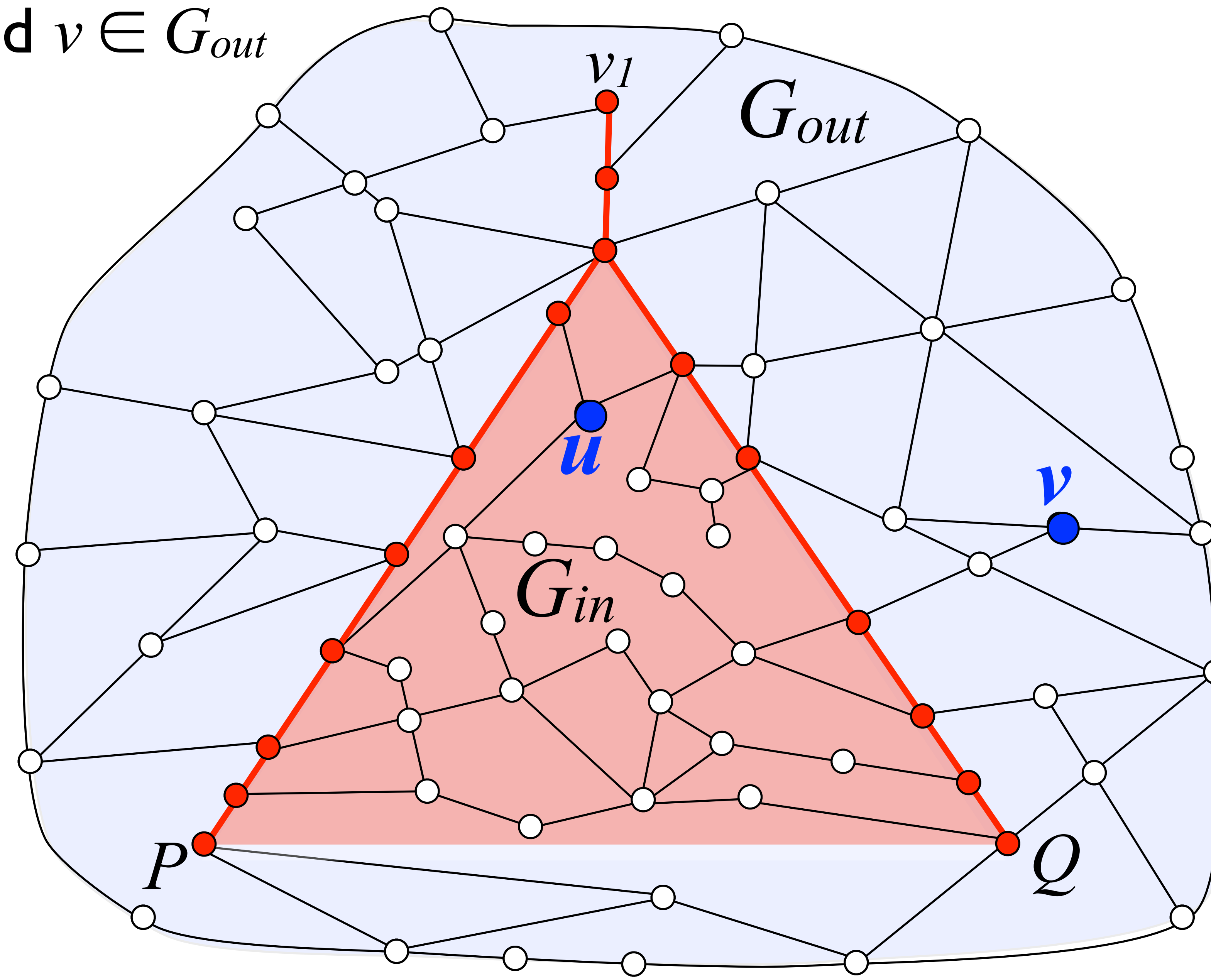


Recursive Algorithm



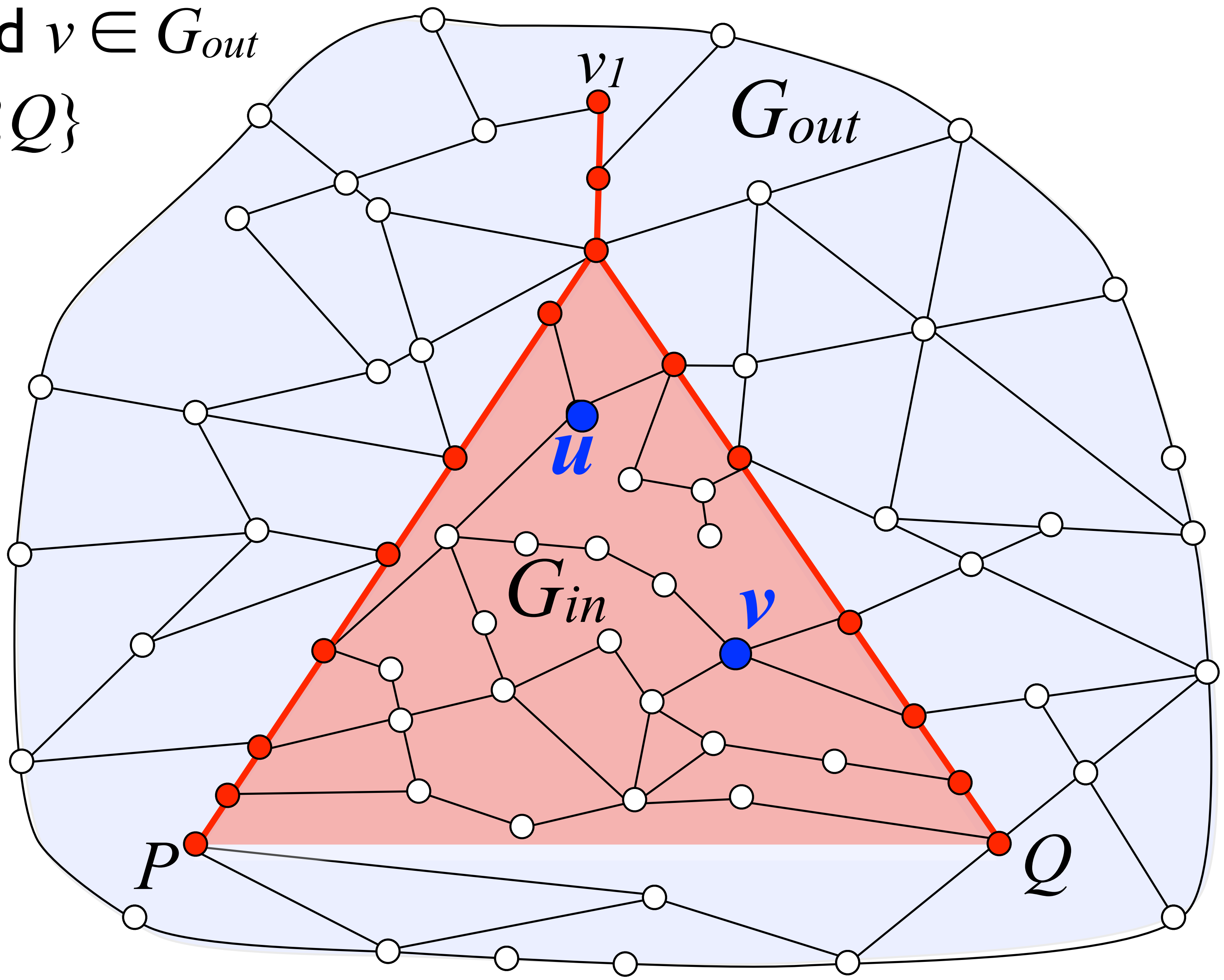
Recursive Algorithm

1. Find furthest pair $u \in G_{in}$ and $v \in G_{out}$



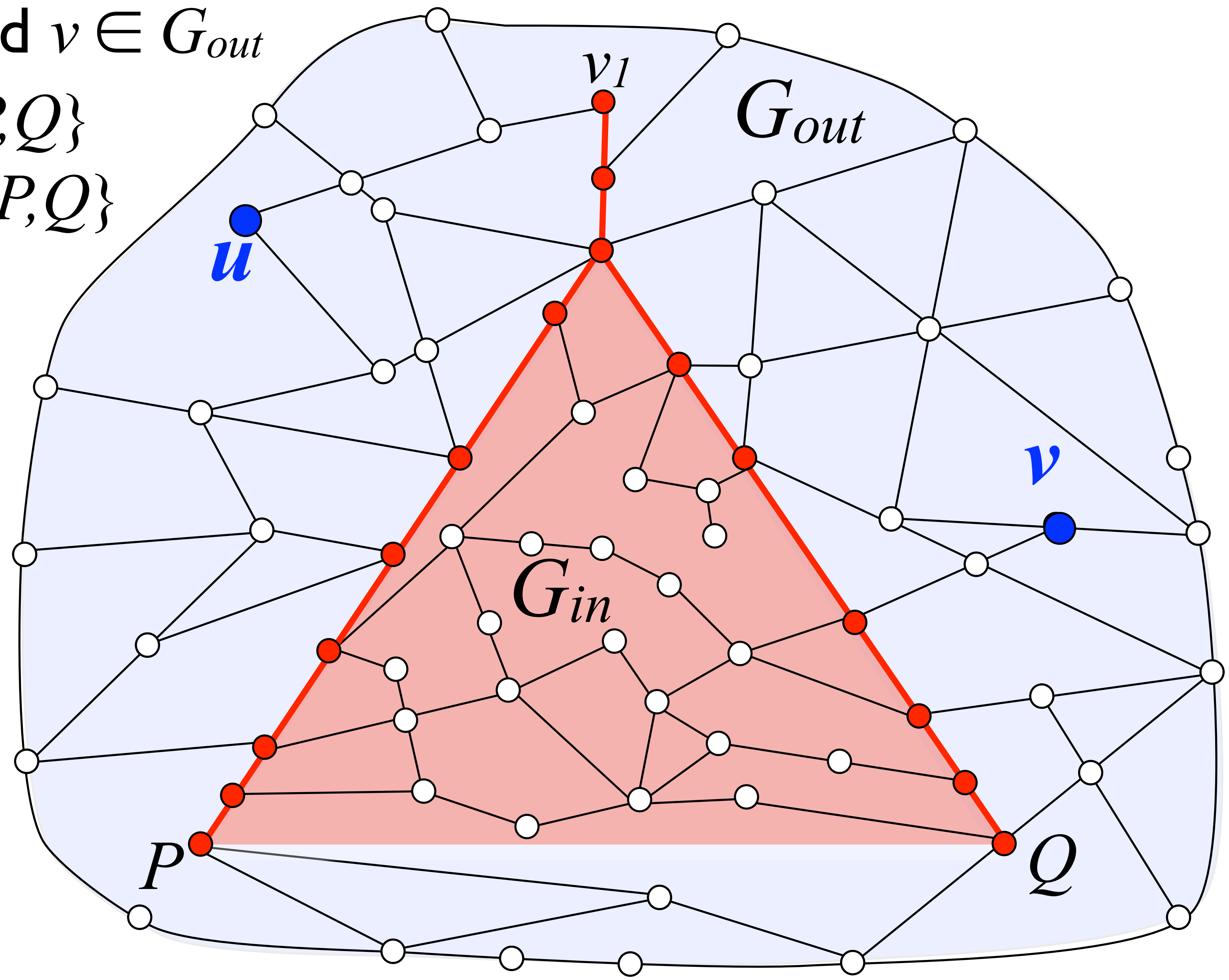
Recursive Algorithm

1. Find furthest pair $u \in G_{in}$ and $v \in G_{out}$
2. Find furthest pair in $G_{in} \setminus \{P,Q\}$



Recursive Algorithm

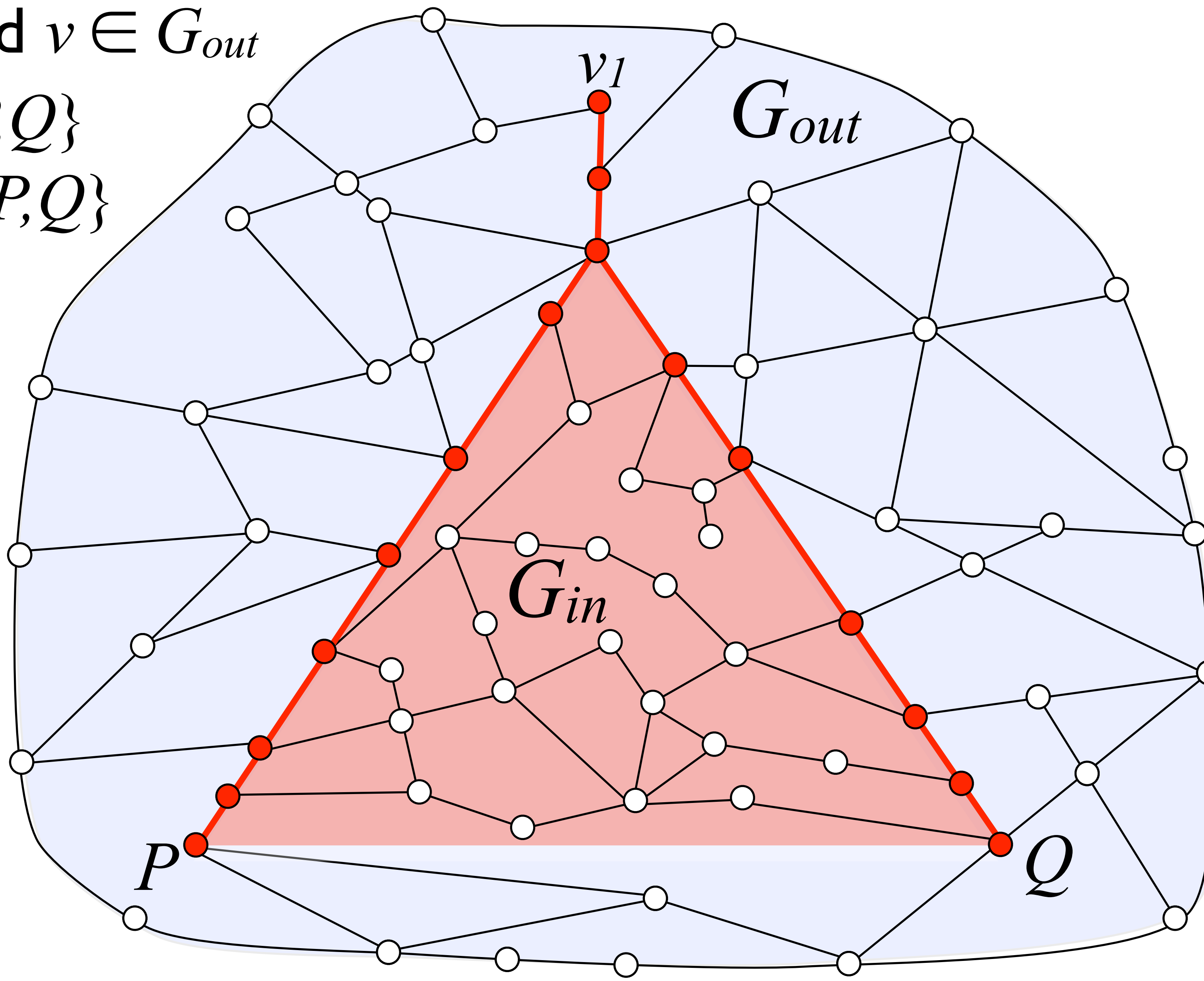
1. Find furthest pair $u \in G_{in}$ and $v \in G_{out}$
2. Find furthest pair in $G_{in} \setminus \{P, Q\}$
3. Find furthest pair in $G_{out} \setminus \{P, Q\}$



Recursive Algorithm

- Mark all nodes

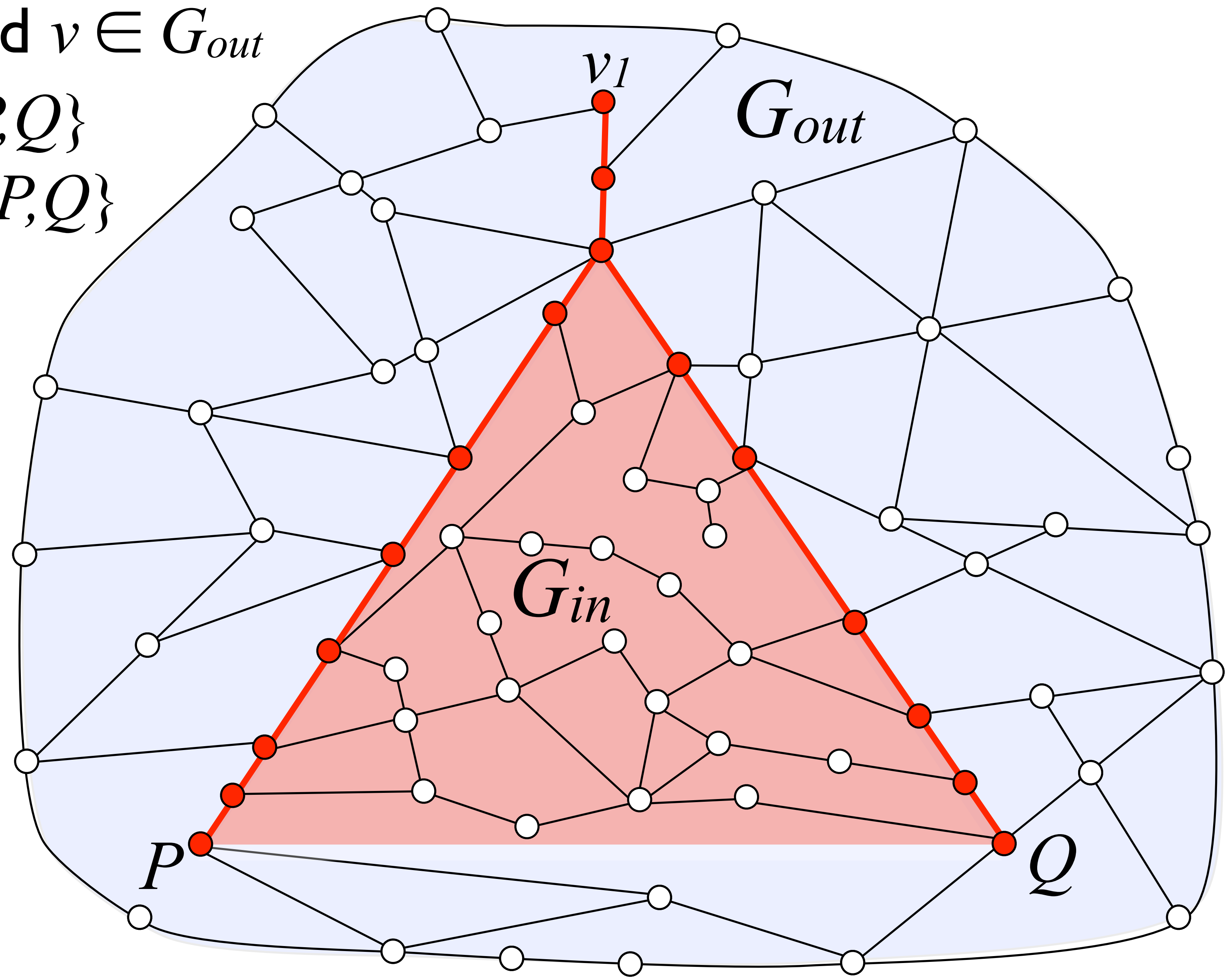
1. Find furthest pair $u \in G_{in}$ and $v \in G_{out}$
2. Find furthest pair in $G_{in} \setminus \{P,Q\}$
3. Find furthest pair in $G_{out} \setminus \{P,Q\}$



Recursive Algorithm

- Mark all nodes
- In $O(n)$ time find x s.t $x \leq$ diameter $\leq 2x$

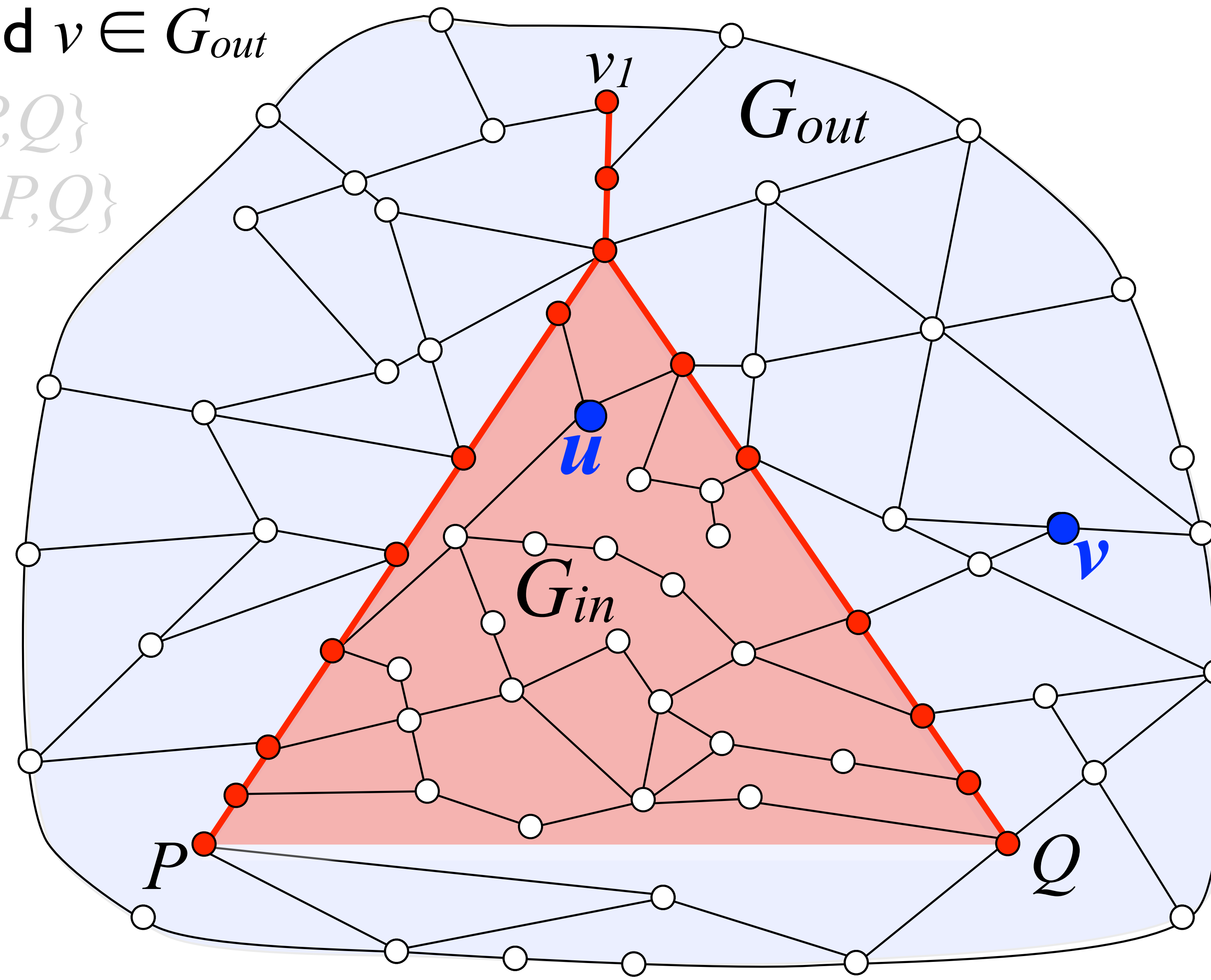
1. Find furthest pair $u \in G_{in}$ and $v \in G_{out}$
2. Find furthest pair in $G_{in} \setminus \{P,Q\}$
3. Find furthest pair in $G_{out} \setminus \{P,Q\}$



Recursive Algorithm

- Mark all nodes
- In $O(n)$ time find x s.t $x \leq$ diameter $\leq 2x$

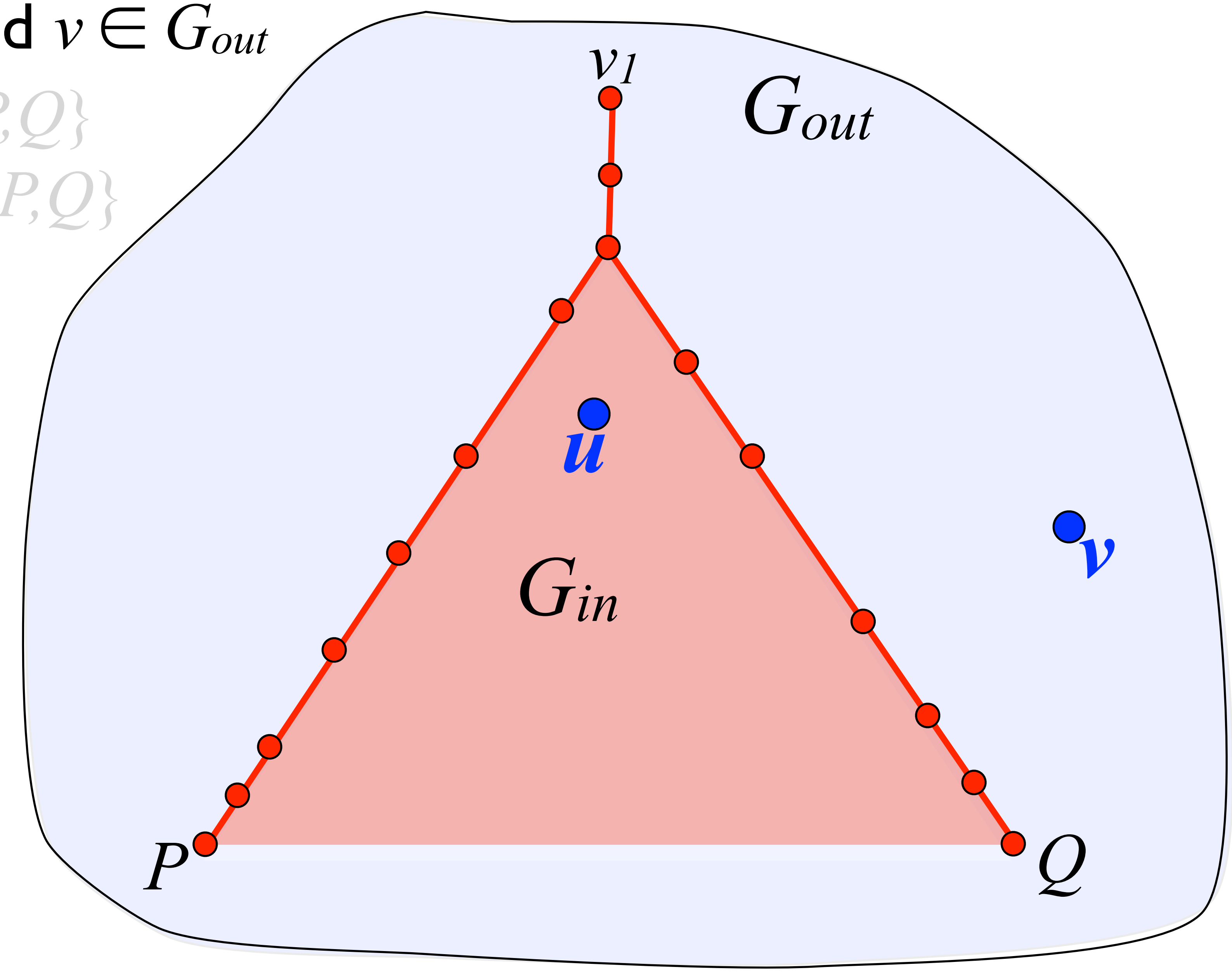
1. Find furthest pair $u \in G_{in}$ and $v \in G_{out}$
2. Find furthest pair in $G_{in} \setminus \{P,Q\}$
3. Find furthest pair in $G_{out} \setminus \{P,Q\}$



Recursive Algorithm

- Mark all nodes
- In $O(n)$ time find x s.t $x \le \text{diameter} \le 2x$

1. Find furthest pair $u \in G_{in}$ and $v \in G_{out}$
2. Find furthest pair in $G_{in} \setminus \{P,Q\}$
3. Find furthest pair in $G_{out} \setminus \{P,Q\}$

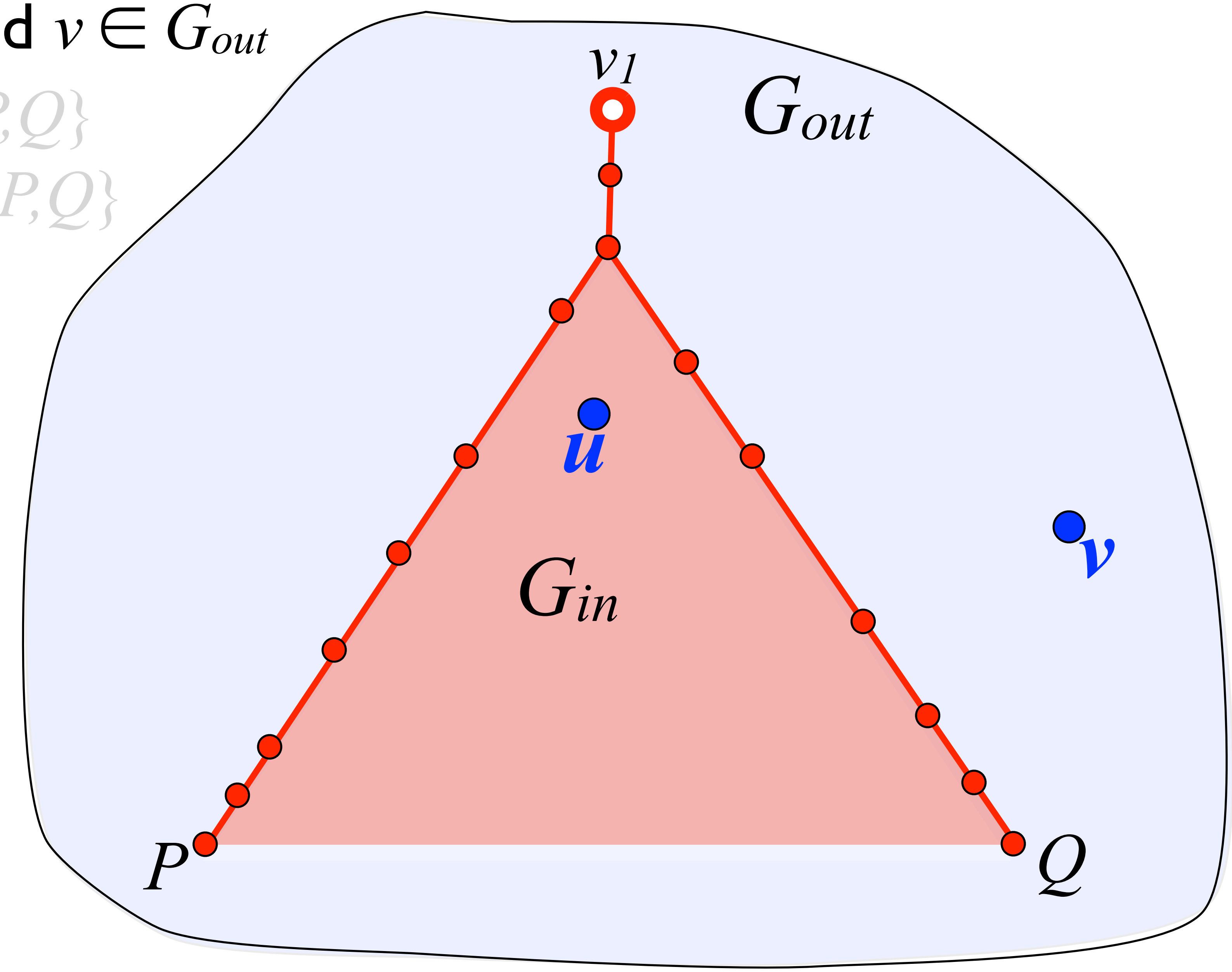


Recursive Algorithm

- Mark all nodes
- In $O(n)$ time find x s.t $x \leq$ diameter $\leq 2x$

1. Find furthest pair $u \in G_{in}$ and $v \in G_{out}$
2. Find furthest pair in $G_{in} \setminus \{P,Q\}$
3. Find furthest pair in $G_{out} \setminus \{P,Q\}$

Choose $16/\varepsilon$ *portals*

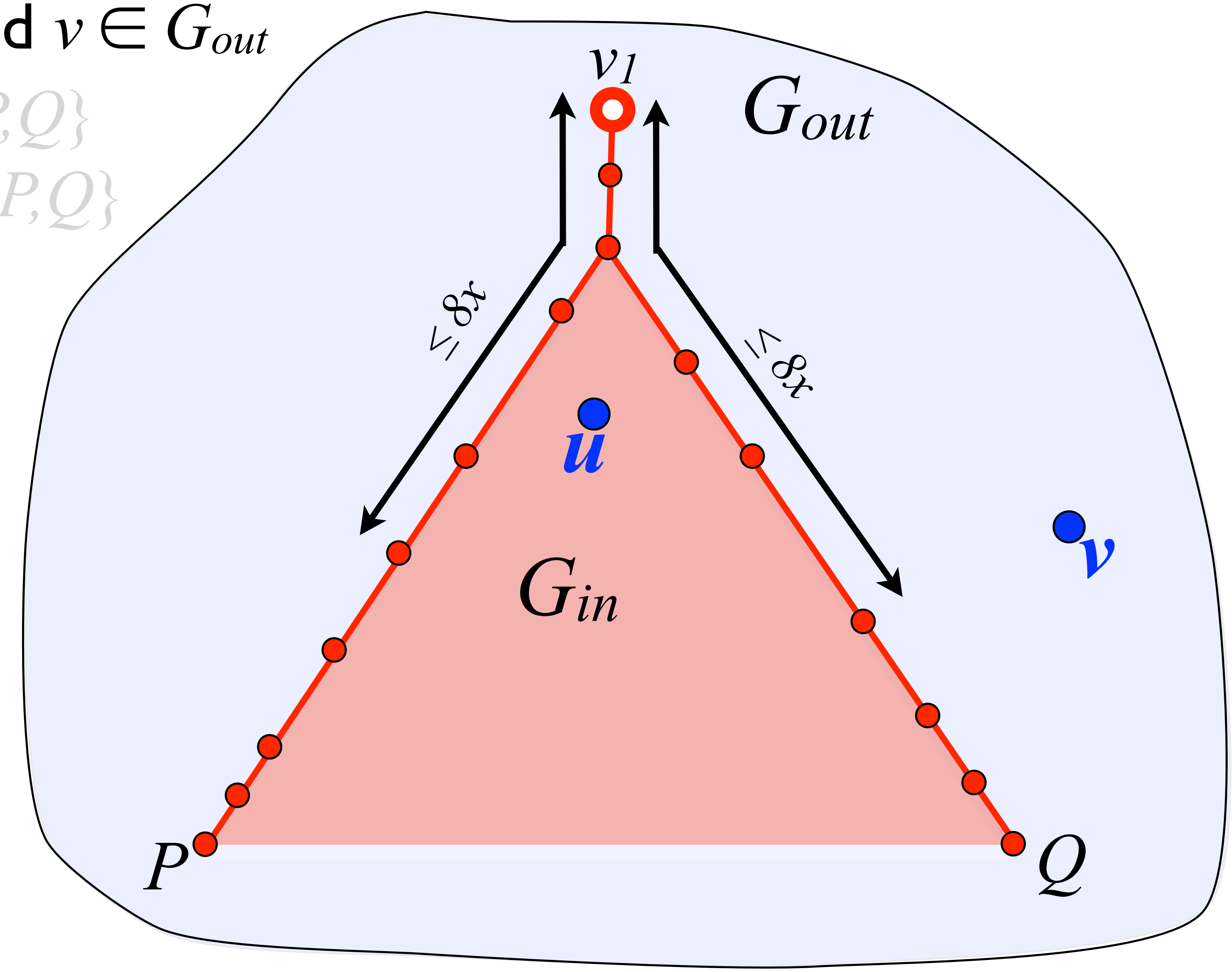


Recursive Algorithm

- Mark all nodes
- In $O(n)$ time find x s.t $x \leq$ diameter $\leq 2x$

1. Find furthest pair $u \in G_{in}$ and $v \in G_{out}$
2. Find furthest pair in $G_{in} \setminus \{P,Q\}$
3. Find furthest pair in $G_{out} \setminus \{P,Q\}$

Choose $16/\varepsilon$ *portals*

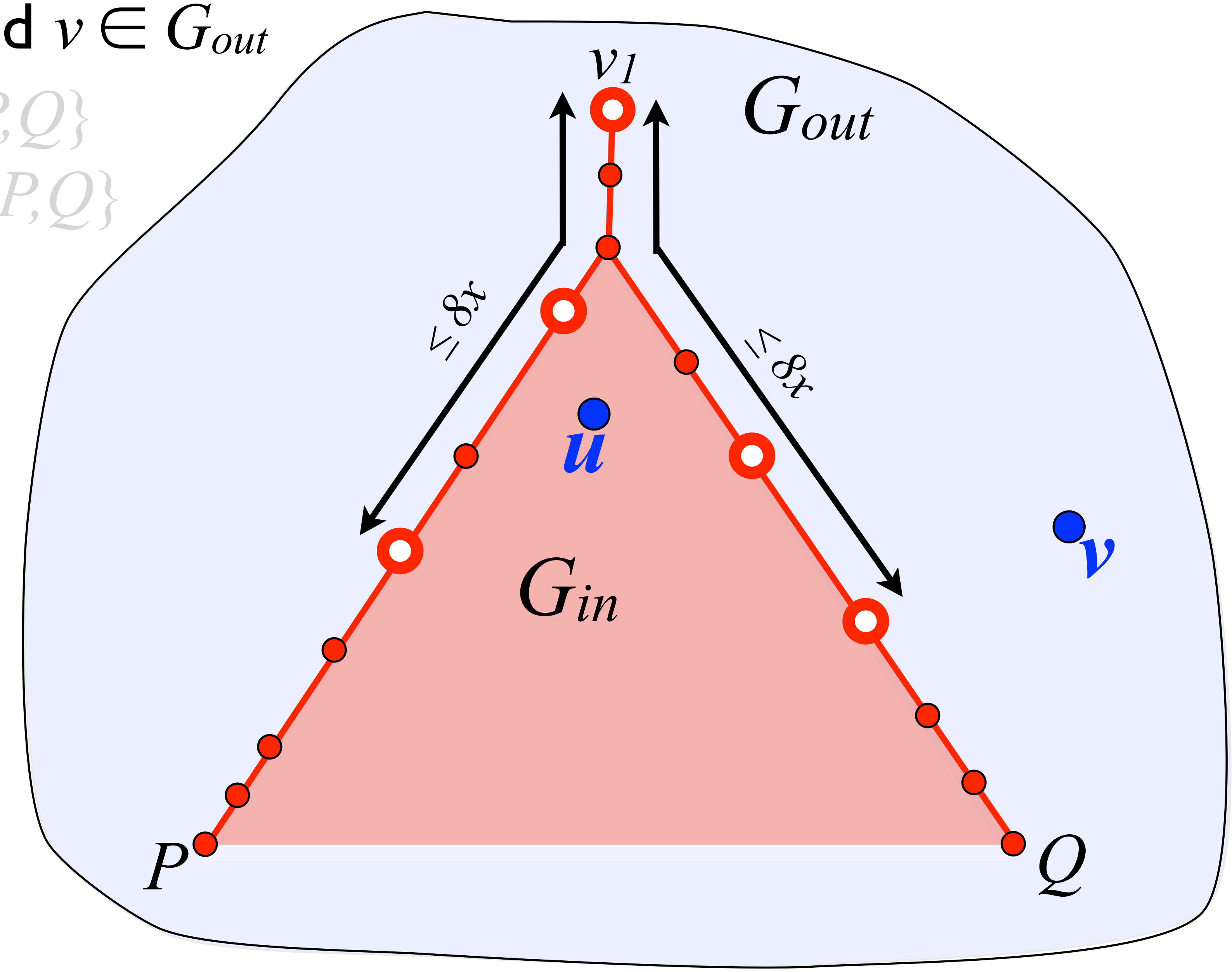


Recursive Algorithm

- Mark all nodes
- In $O(n)$ time find x s.t $x \leq \text{diameter} \leq 2x$

1. Find furthest pair $u \in G_{in}$ and $v \in G_{out}$
2. Find furthest pair in $G_{in} \setminus \{P,Q\}$
3. Find furthest pair in $G_{out} \setminus \{P,Q\}$

Choose $16/\varepsilon$ *portals*

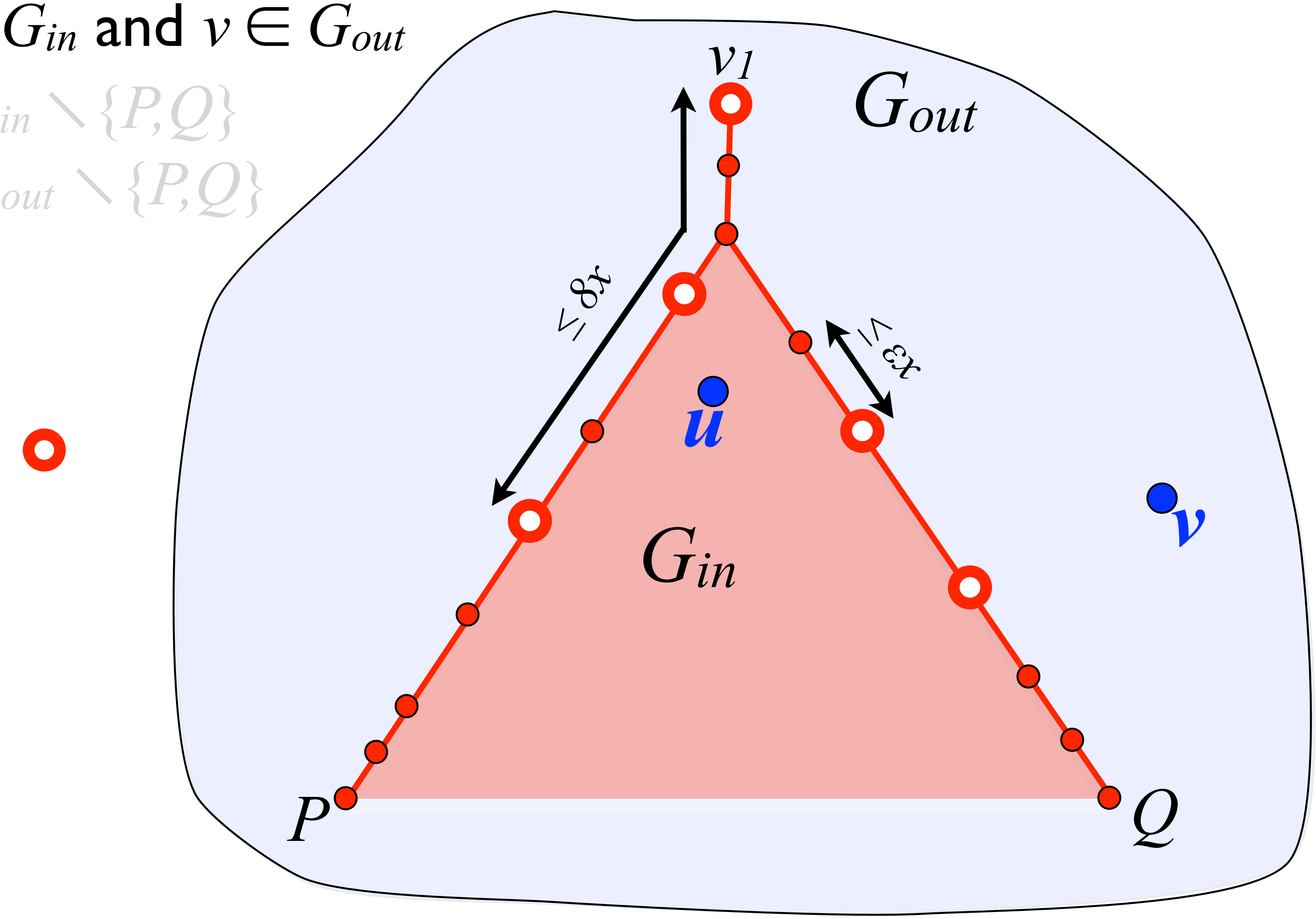


Recursive Algorithm

- Mark all nodes
- In $O(n)$ time find x s.t $x \leq \text{diameter} \leq 2x$

1. Find furthest pair $u \in G_{in}$ and $v \in G_{out}$
2. Find furthest pair in $G_{in} \setminus \{P,Q\}$
3. Find furthest pair in $G_{out} \setminus \{P,Q\}$

Choose $16/\varepsilon$ *portals*

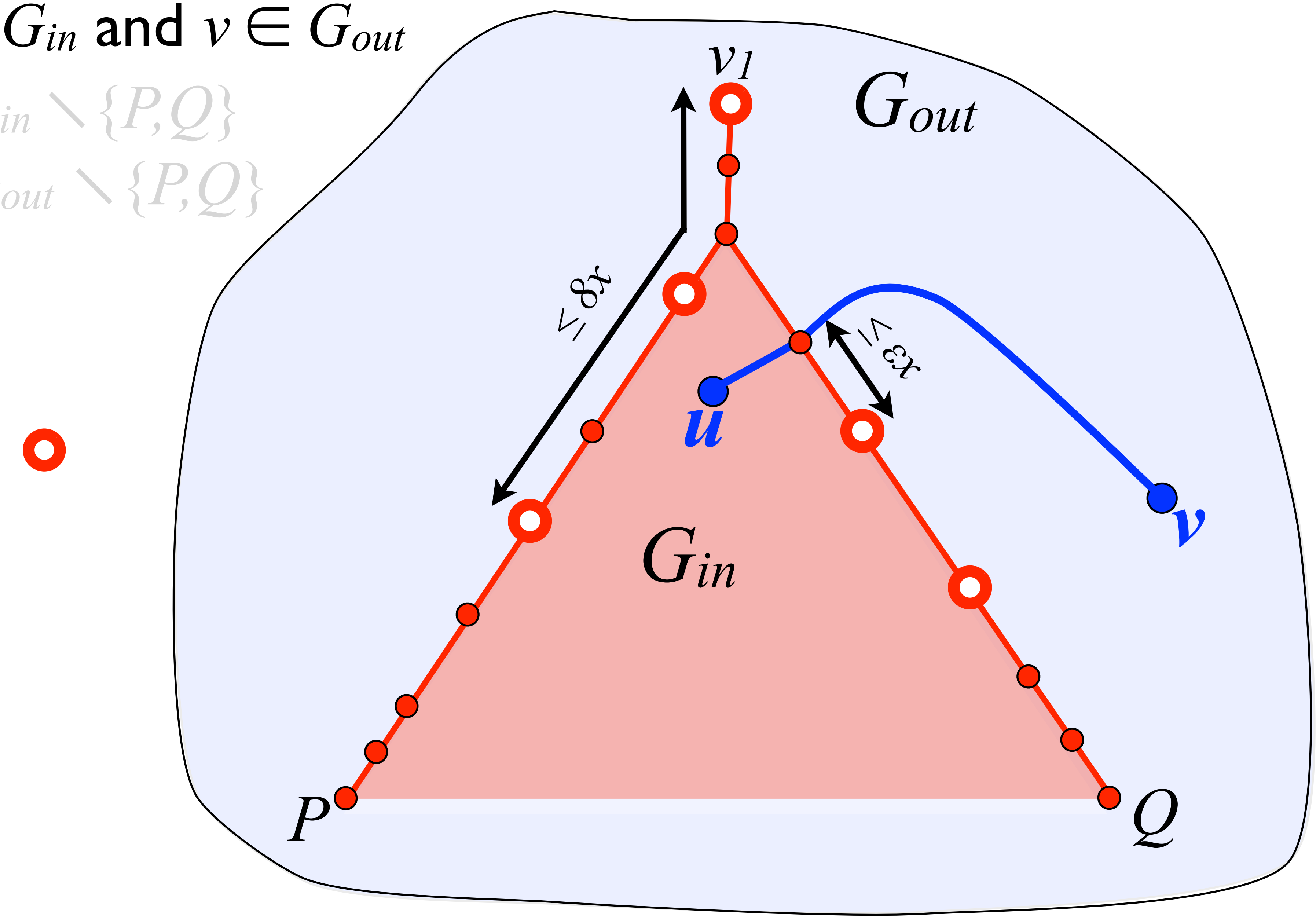


Recursive Algorithm

- Mark all nodes
- In $O(n)$ time find x s.t $x \leq$ diameter $\leq 2x$

1. Find furthest pair $u \in G_{in}$ and $v \in G_{out}$
2. Find furthest pair in $G_{in} \setminus \{P,Q\}$
3. Find furthest pair in $G_{out} \setminus \{P,Q\}$

Choose $16/\varepsilon$ *portals*

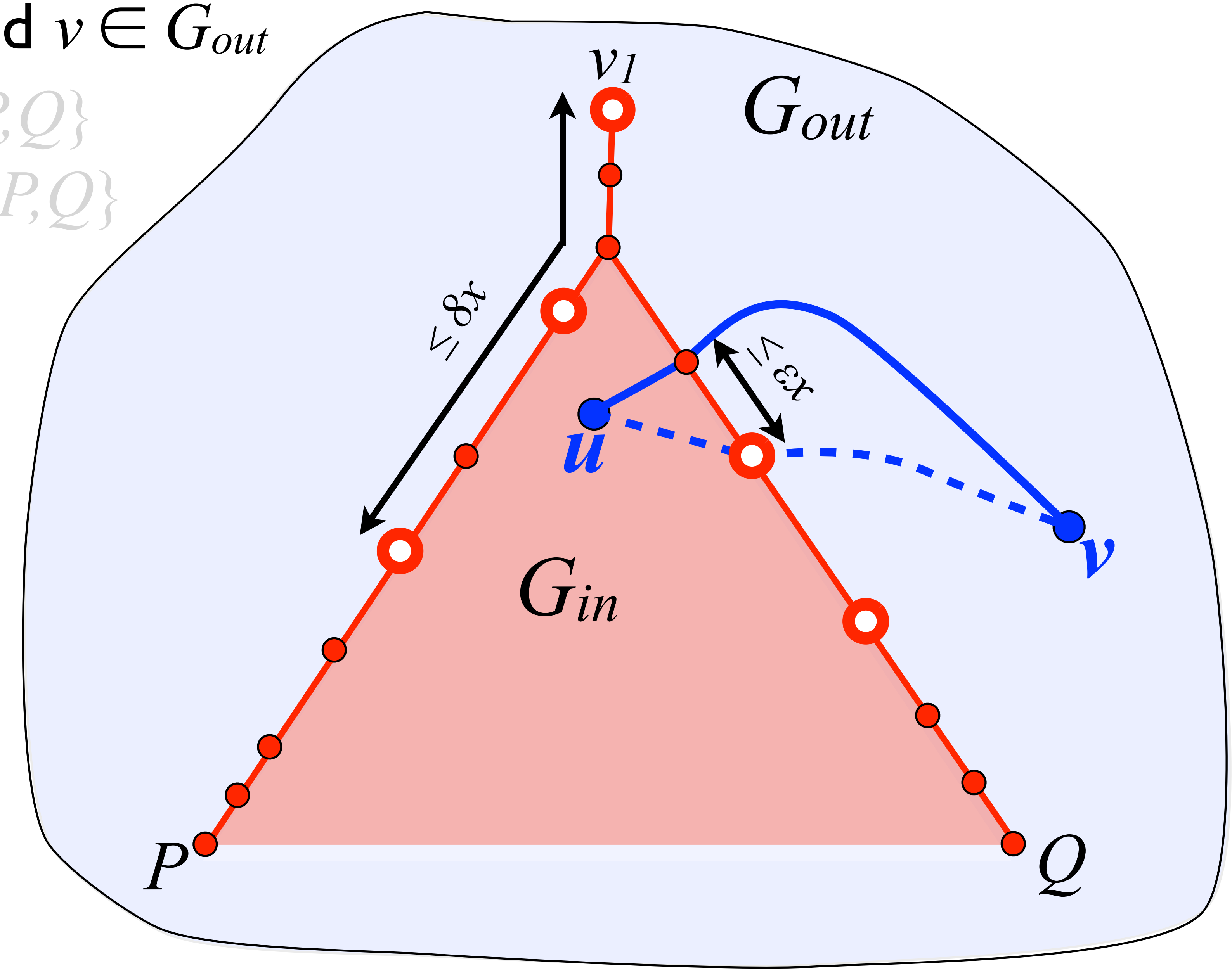


Recursive Algorithm

- Mark all nodes
- In $O(n)$ time find x s.t $x \leq$ diameter $\leq 2x$

1. Find furthest pair $u \in G_{in}$ and $v \in G_{out}$
2. Find furthest pair in $G_{in} \setminus \{P,Q\}$
3. Find furthest pair in $G_{out} \setminus \{P,Q\}$

Choose $16/\varepsilon$ *portals*



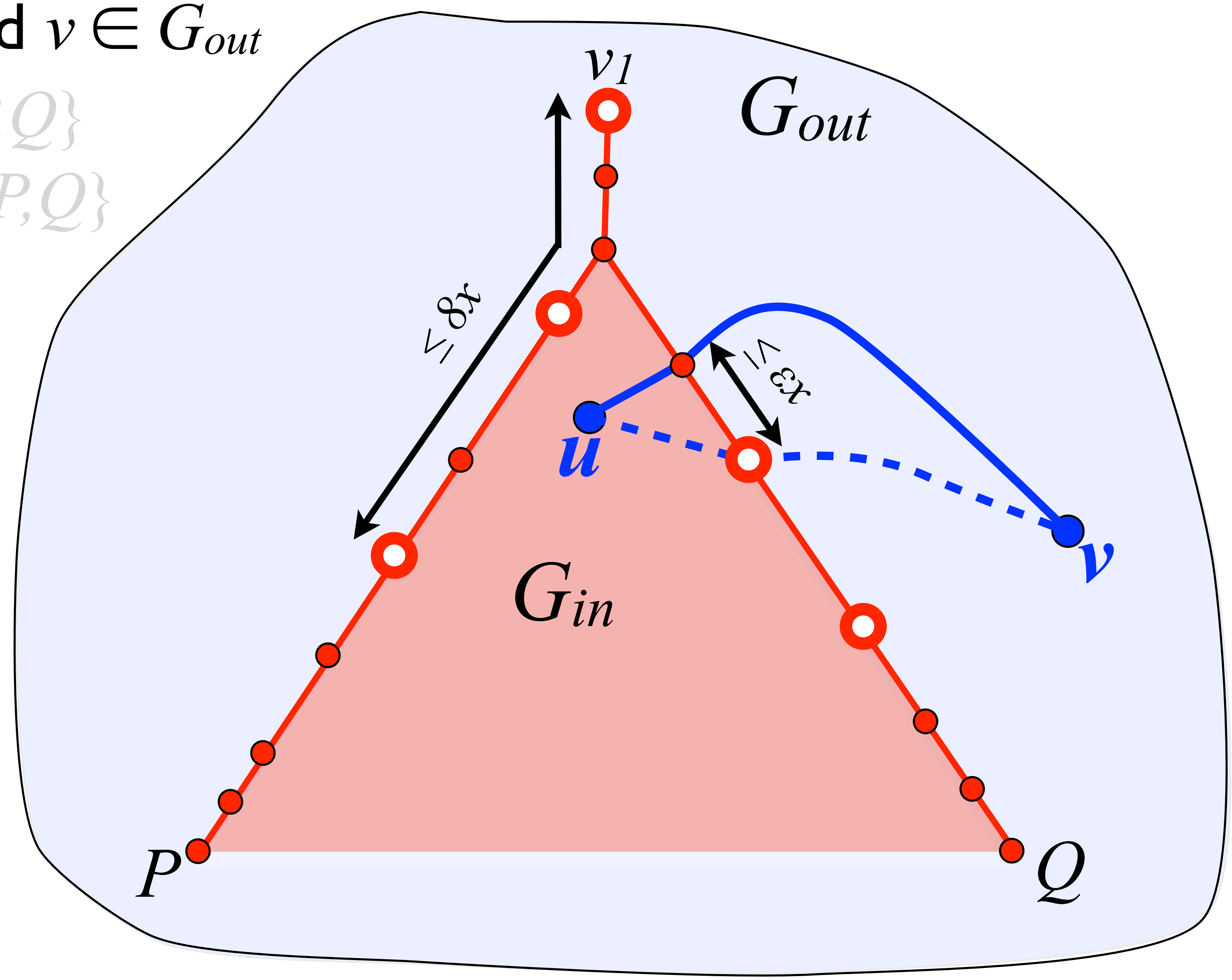
Recursive Algorithm

- Mark all nodes
- In $O(n)$ time find x s.t $x \leq$ diameter $\leq 2x$

1. Find furthest pair $u \in G_{in}$ and $v \in G_{out}$
2. Find furthest pair in $G_{in} \setminus \{P,Q\}$
3. Find furthest pair in $G_{out} \setminus \{P,Q\}$

Choose $16/\varepsilon$ *portals*

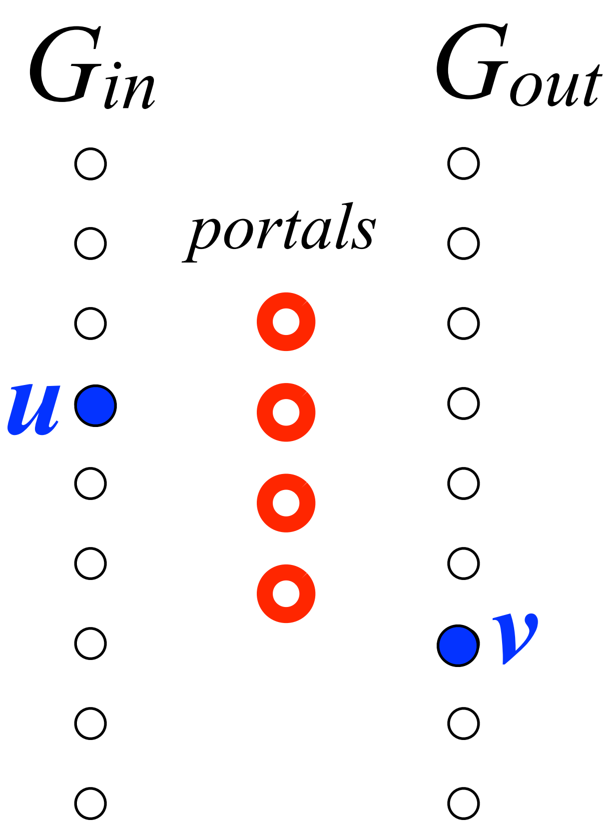
Lemma: a shortest ***u***-to-***v*** path does not cross below the $8x$ prefix

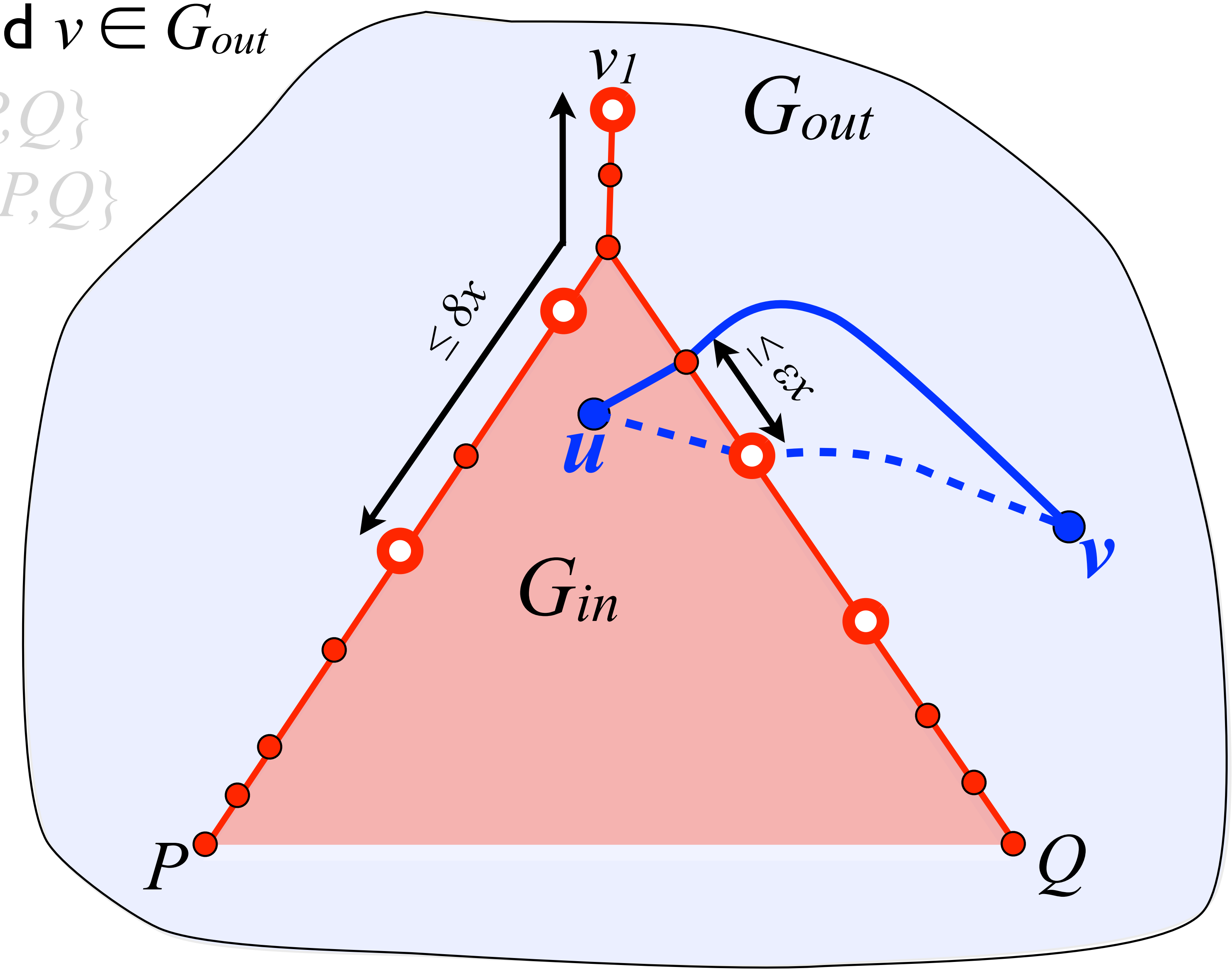


Recursive Algorithm

- Mark all nodes
- In $O(n)$ time find x s.t $x \leq \text{diameter} \leq 2x$

1. Find furthest pair $u \in G_{in}$ and $v \in G_{out}$
2. Find furthest pair in $G_{in} \setminus \{P,Q\}$
3. Find furthest pair in $G_{out} \setminus \{P,Q\}$

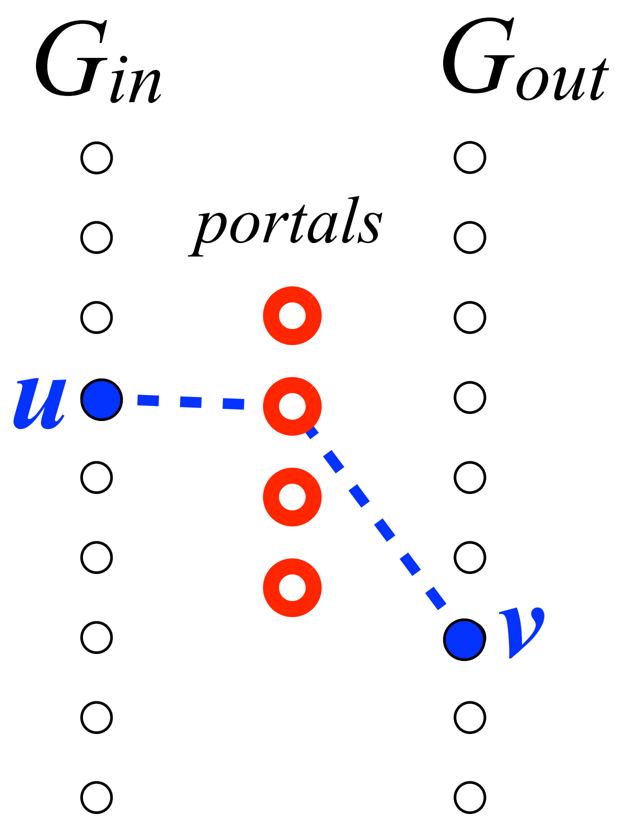


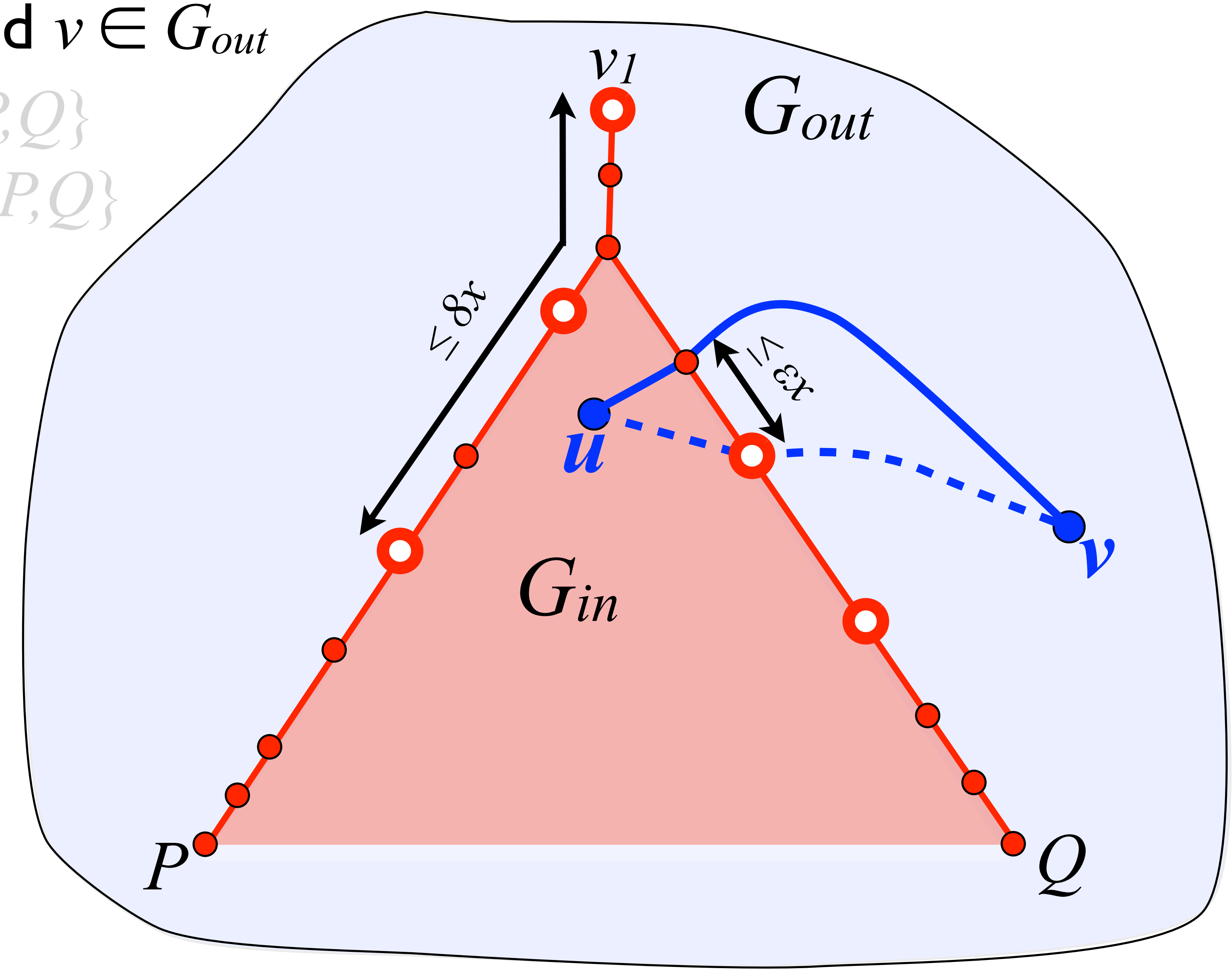


Recursive Algorithm

- Mark all nodes
- In $O(n)$ time find x s.t $x \leq$ diameter $\leq 2x$

1. Find furthest pair $u \in G_{in}$ and $v \in G_{out}$
2. Find furthest pair in $G_{in} \setminus \{P,Q\}$
3. Find furthest pair in $G_{out} \setminus \{P,Q\}$

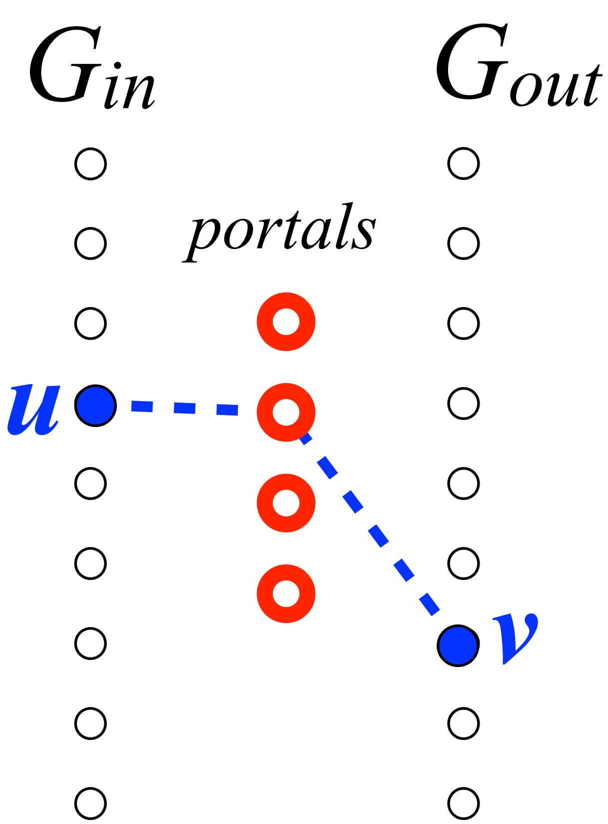




Recursive Algorithm

- Mark all nodes
- In $O(n)$ time find x s.t $x \leq \text{diameter} \leq 2x$

1. Find furthest pair $u \in G_{in}$ and $v \in G_{out}$
2. Find furthest pair in $G_{in} \setminus \{P,Q\}$
3. Find furthest pair in $G_{out} \setminus \{P,Q\}$

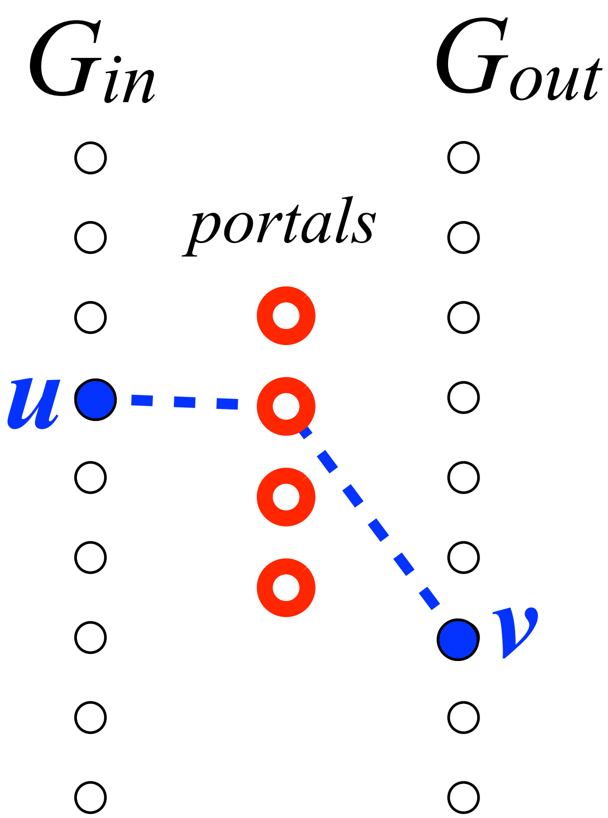


Lemma: we can round the edge lengths to be in $\{1,2,\ldots,1/\varepsilon\}$

Recursive Algorithm

- Mark all nodes
- In $O(n)$ time find x s.t $x \leq \text{diameter} \leq 2x$

1. Find furthest pair $u \in G_{in}$ and $v \in G_{out}$
2. Find furthest pair in $G_{in} \setminus \{P,Q\}$
3. Find furthest pair in $G_{out} \setminus \{P,Q\}$



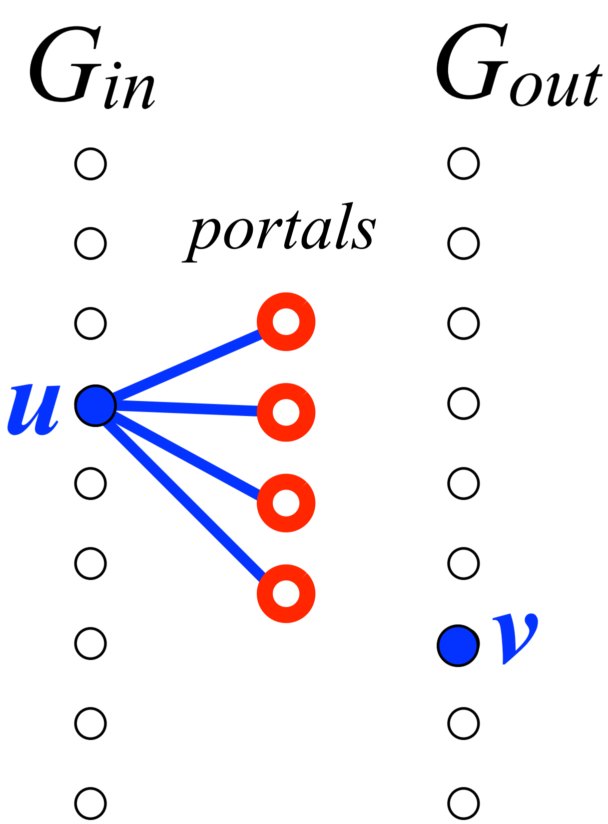
Lemma: we can round the edge lengths to be in $\{1,2,\ldots,1/\varepsilon\}$

proof: Use $x \leq \text{diameter} \leq 2x$

Recursive Algorithm

- Mark all nodes
- In $O(n)$ time find x s.t $x \leq$ diameter $\leq 2x$

1. Find furthest pair $u \in G_{in}$ and $v \in G_{out}$
2. Find furthest pair in $G_{in} \setminus \{P, Q\}$
3. Find furthest pair in $G_{out} \setminus \{P, Q\}$



Lemma: we can round the edge lengths to be in $\{1,2,\ldots,1/\varepsilon\}$

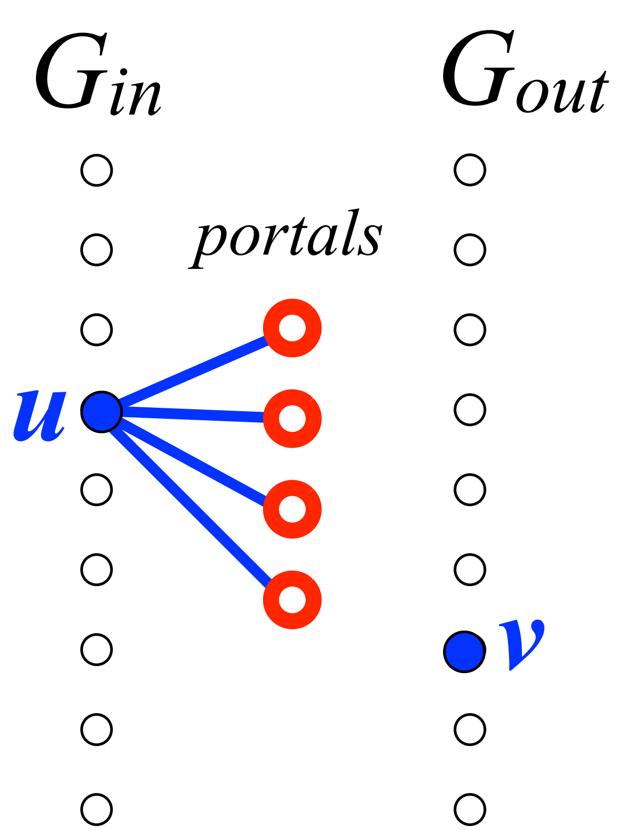
proof: Use $x \leq$ diameter $\leq 2x$

Lemma: after rounding we can find the exact diameter of the tripartite graph in time $O(2^{O(1/\varepsilon)} \cdot n)$

Recursive Algorithm

- Mark all nodes
- In $O(n)$ time find x s.t $x \leq \text{diameter} \leq 2x$

1. Find furthest pair $u \in G_{in}$ and $v \in G_{out}$
2. Find furthest pair in $G_{in} \setminus \{P,Q\}$
3. Find furthest pair in $G_{out} \setminus \{P,Q\}$



Lemma: we can round the edge lengths to be in $\{1,2,\ldots,1/\varepsilon\}$

proof: Use $x \leq \text{diameter} \leq 2x$

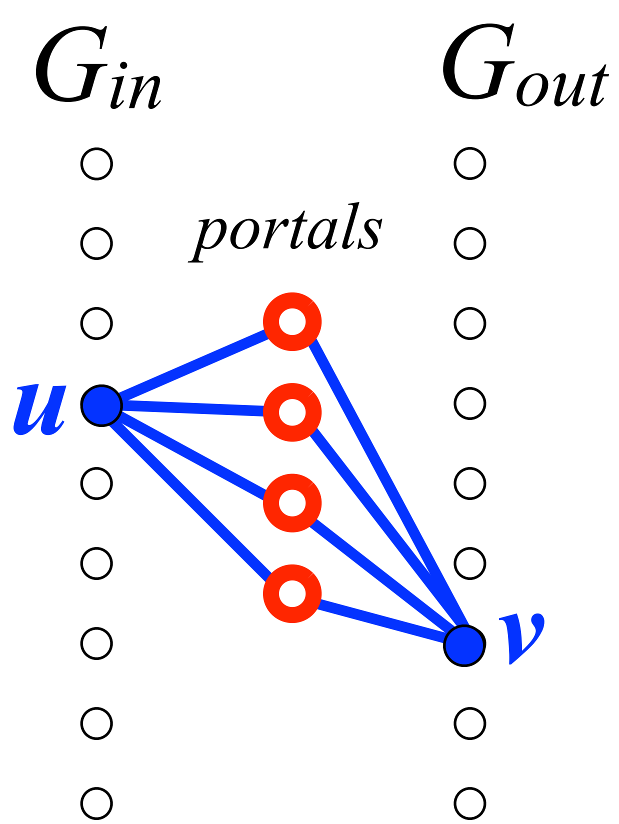
Lemma: after rounding we can find the exact diameter of the tripartite graph in time $O(2^{O(1/\varepsilon)} \cdot n)$

proof: $2^{O(1/\varepsilon)}$ config. of what u can "see"

Recursive Algorithm

- Mark all nodes
- In $O(n)$ time find x s.t $x \le$ diameter $\le 2x$

1. Find furthest pair $u \in G_{in}$ and $v \in G_{out}$
2. Find furthest pair in $G_{in} \setminus \{P,Q\}$
3. Find furthest pair in $G_{out} \setminus \{P,Q\}$



Lemma: we can round the edge lengths to be in $\{1,2,\ldots,1/\varepsilon\}$

proof: Use $x \le$ diameter $\le 2x$

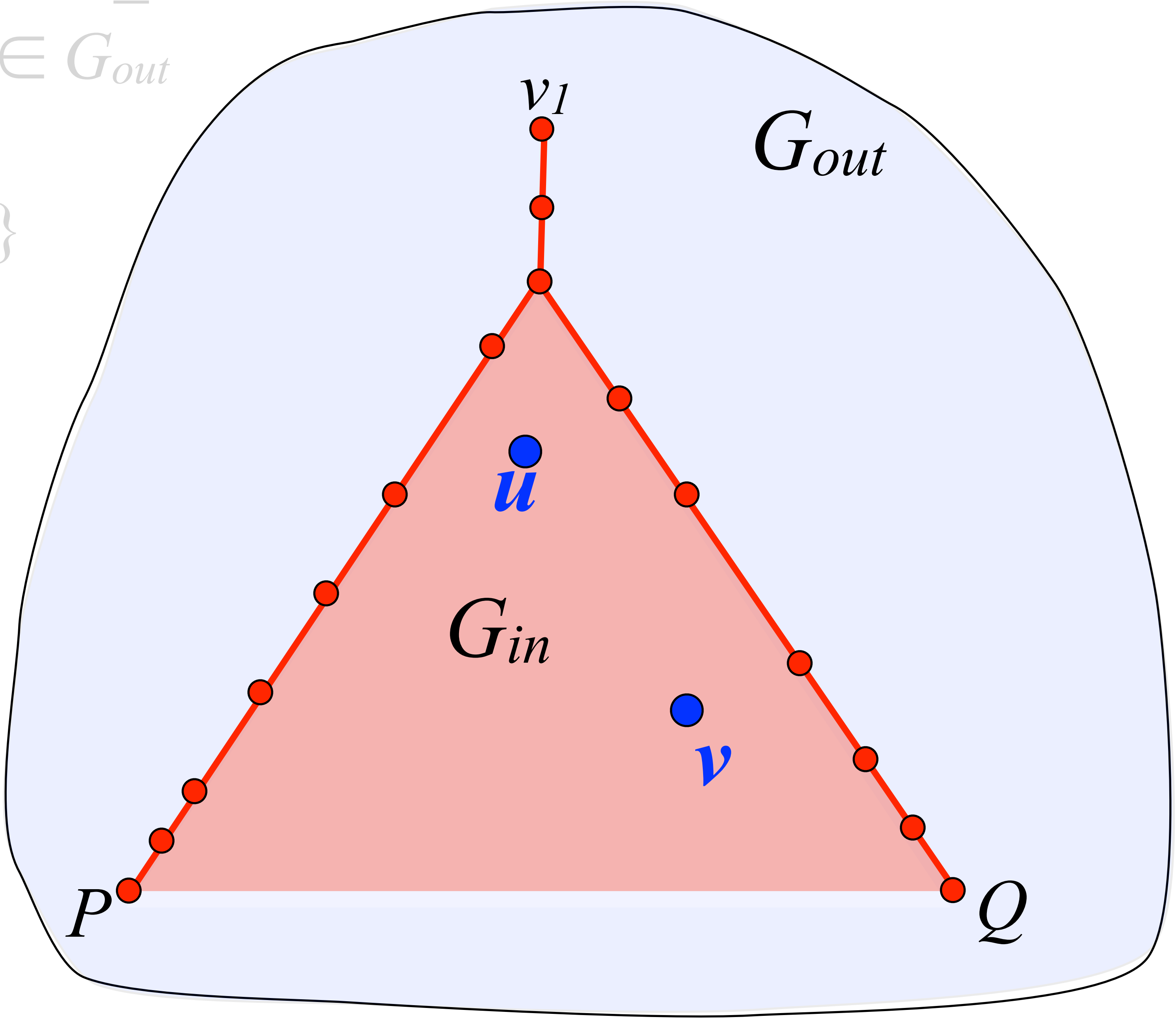
Lemma: after rounding we can find the exact diameter of the tripartite graph in time $O(2^{O(1/\varepsilon)} \cdot n)$

proof: $2^{O(1/\varepsilon)}$ config. of what u can "see"

Recursive Algorithm

- Mark all nodes
- In $O(n)$ time find x s.t $x \leq$ diameter $\leq 2x$

1. Find furthest pair $u \in G_{in}$ and $v \in G_{out}$
2. Find furthest pair in $G_{in} \setminus \{P,Q\}$
3. Find furthest pair in $G_{out} \setminus \{P,Q\}$

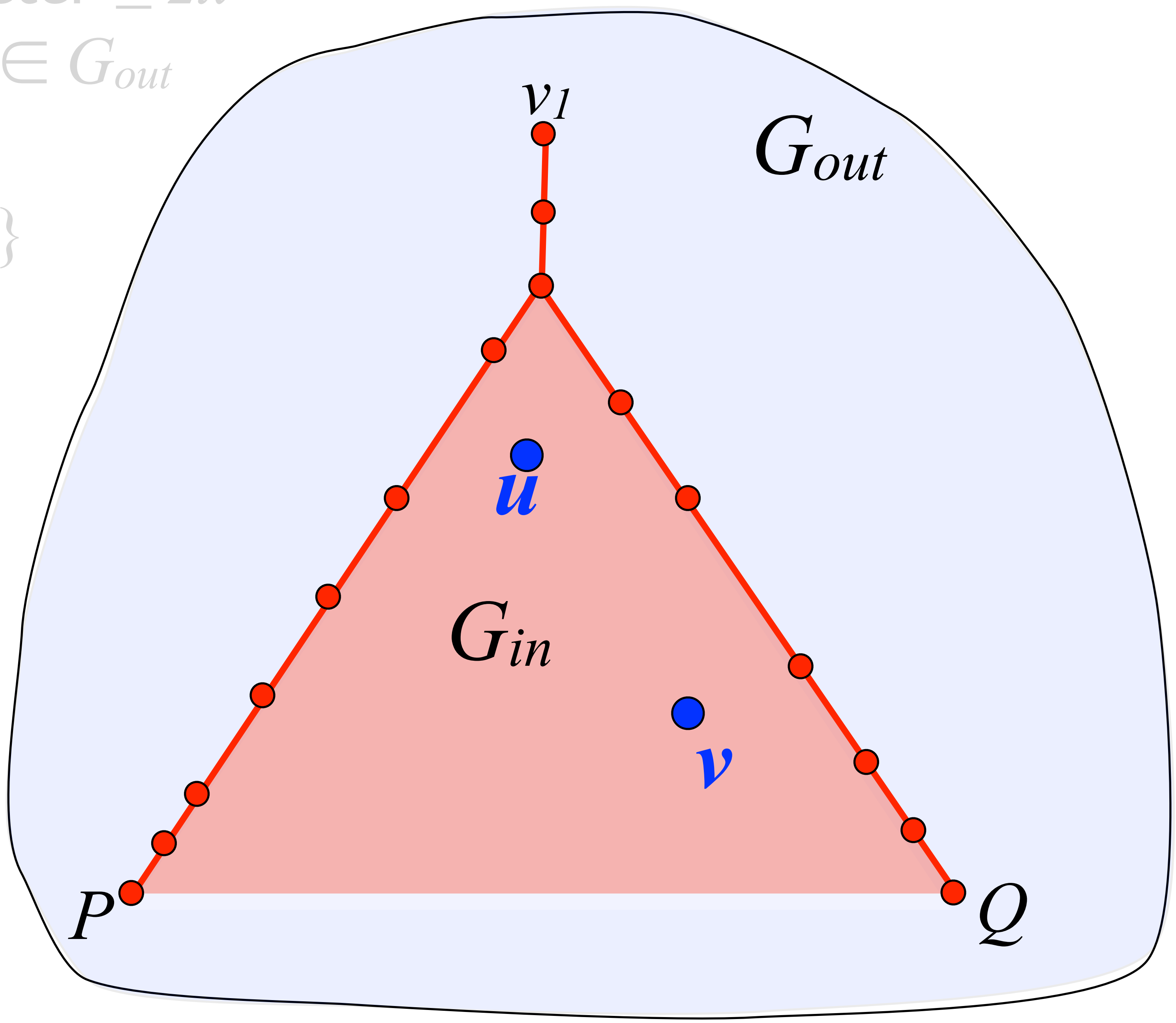


Recursive Algorithm

- Mark all nodes
- In $O(n)$ time find x s.t $x \leq \text{diameter} \leq 2x$

1. Find furthest pair $u \in G_{in}$ and $v \in G_{out}$
2. Find furthest pair in $G_{in} \setminus \{P,Q\}$
3. Find furthest pair in $G_{out} \setminus \{P,Q\}$

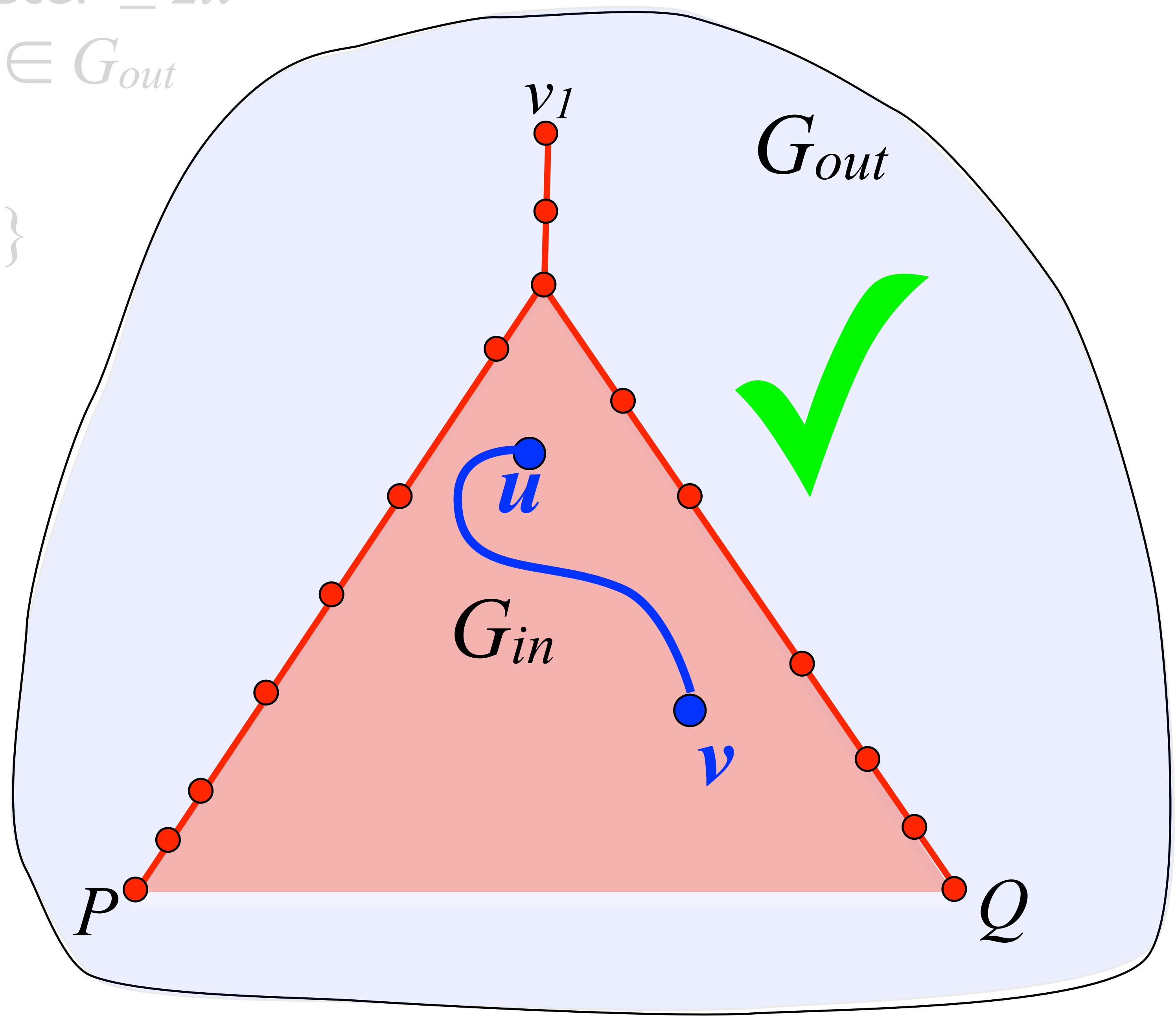
First unmark all nodes of $\{P,Q\}$



Recursive Algorithm

- Mark all nodes
- In $O(n)$ time find x s.t $x \leq \text{diameter} \leq 2x$

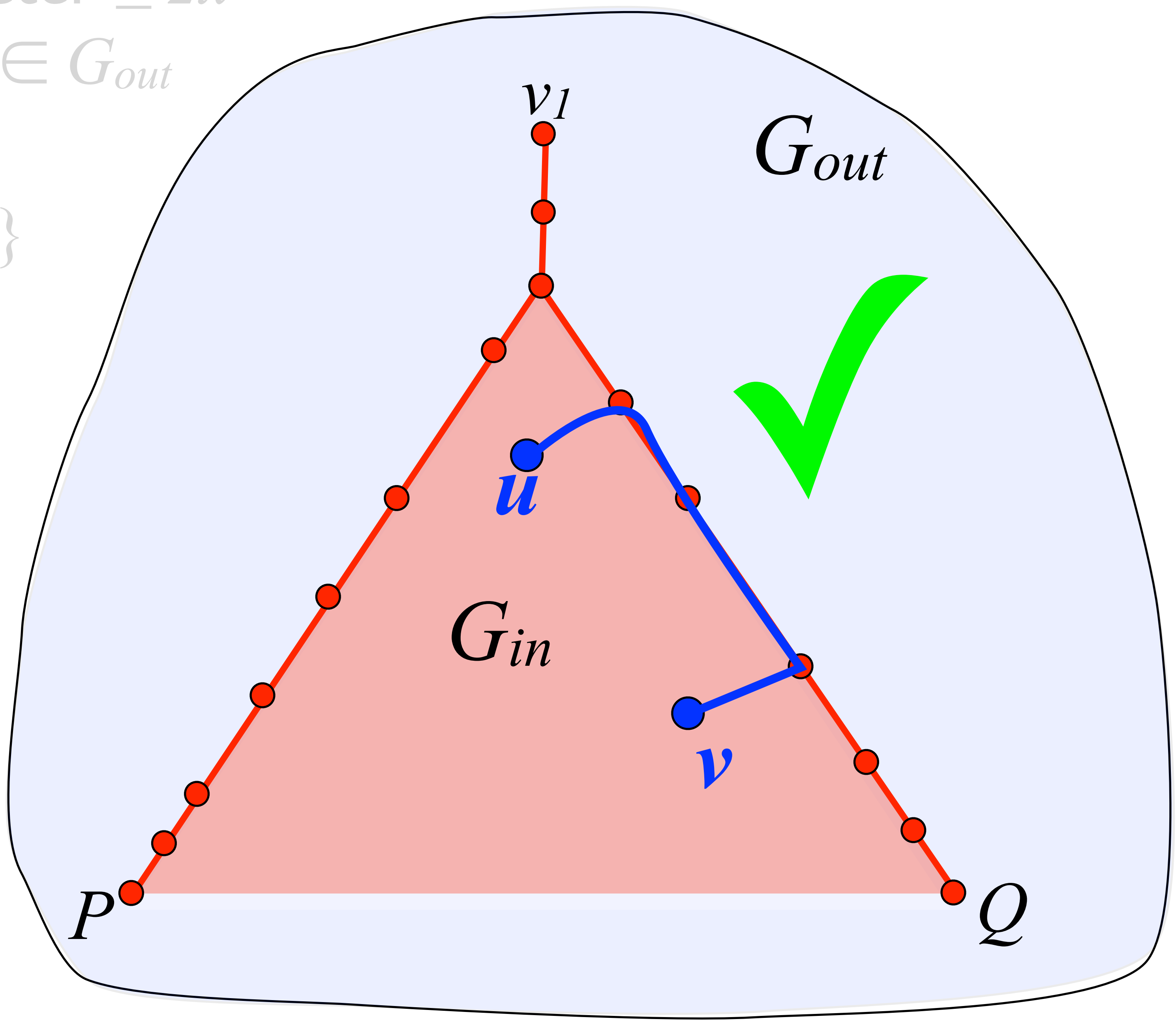
1. Find furthest pair $u \in G_{in}$ and $v \in G_{out}$
2. Find furthest pair in $G_{in} \setminus \{P,Q\}$
3. Find furthest pair in $G_{out} \setminus \{P,Q\}$



Recursive Algorithm

- Mark all nodes
- In $O(n)$ time find x s.t $x \leq \text{diameter} \leq 2x$

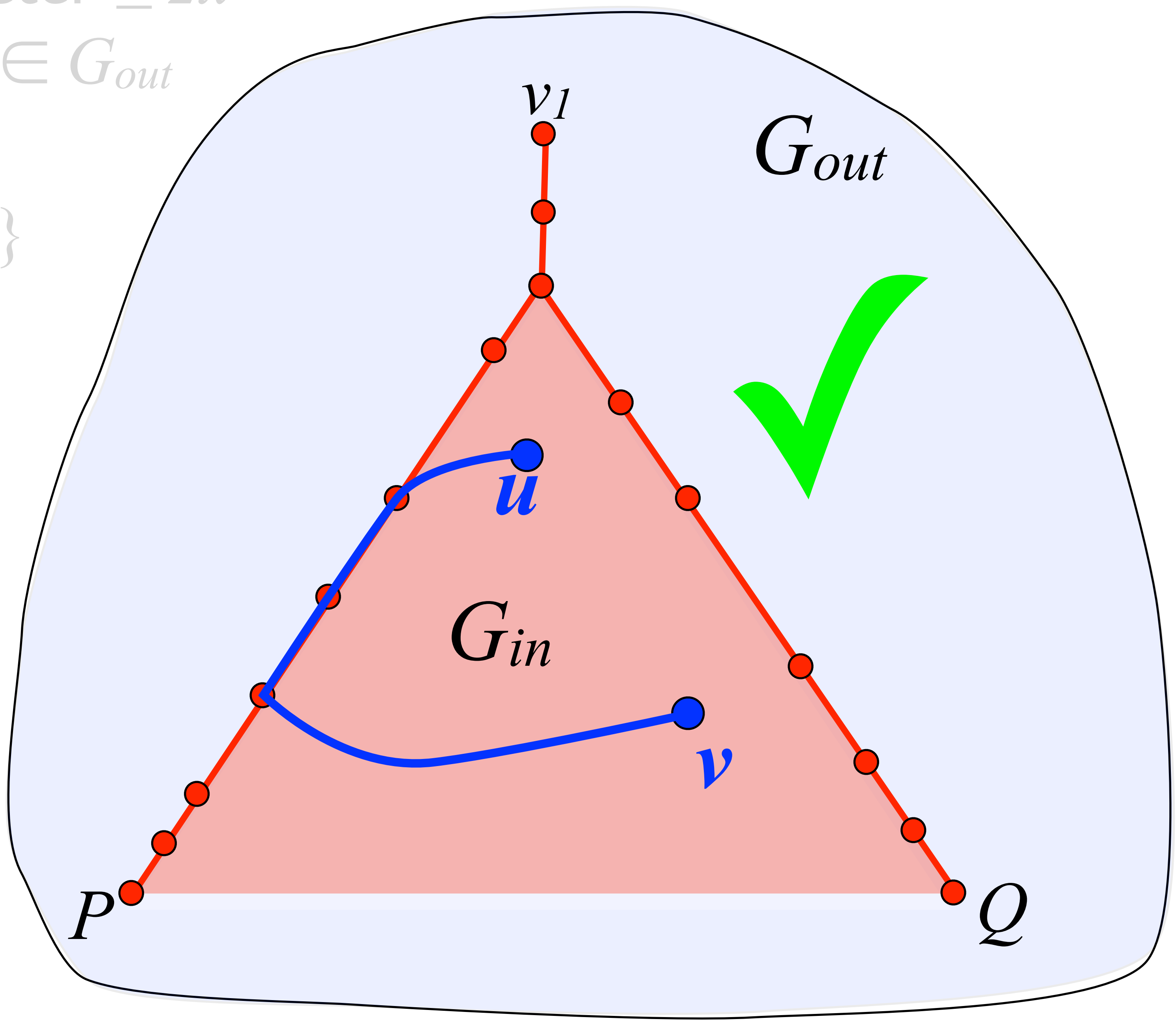
1. Find furthest pair $u \in G_{in}$ and $v \in G_{out}$
2. Find furthest pair in $G_{in} \setminus \{P,Q\}$
3. Find furthest pair in $G_{out} \setminus \{P,Q\}$



Recursive Algorithm

- Mark all nodes
- In $O(n)$ time find x s.t $x \leq$ diameter $\leq 2x$

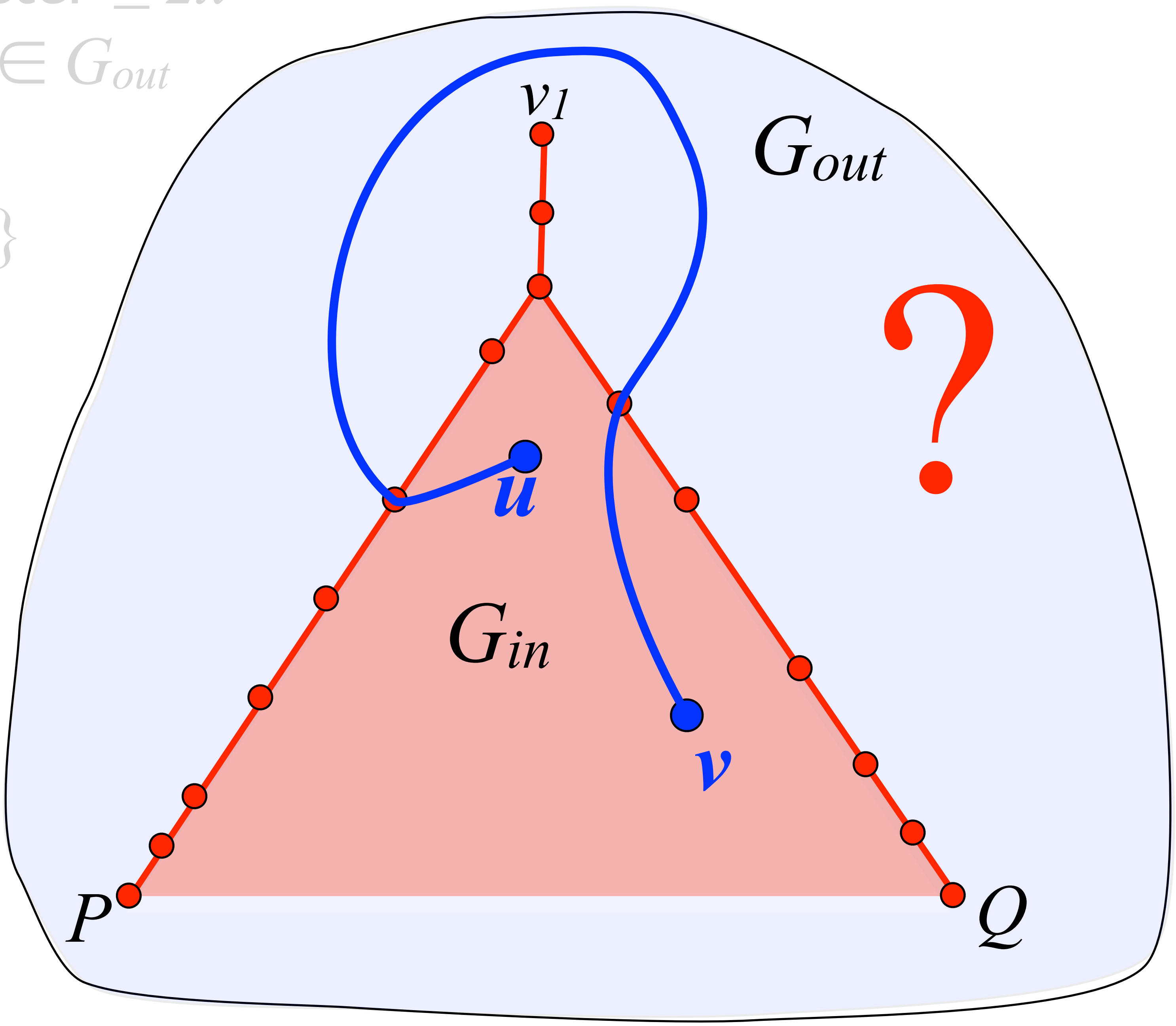
1. Find furthest pair $u \in G_{in}$ and $v \in G_{out}$
2. Find furthest pair in $G_{in} \setminus \{P,Q\}$
3. Find furthest pair in $G_{out} \setminus \{P,Q\}$



Recursive Algorithm

- Mark all nodes
- In $O(n)$ time find x s.t $x \leq$ diameter $\leq 2x$

1. Find furthest pair $u \in G_{in}$ and $v \in G_{out}$
2. Find furthest pair in $G_{in} \setminus \{P,Q\}$
3. Find furthest pair in $G_{out} \setminus \{P,Q\}$



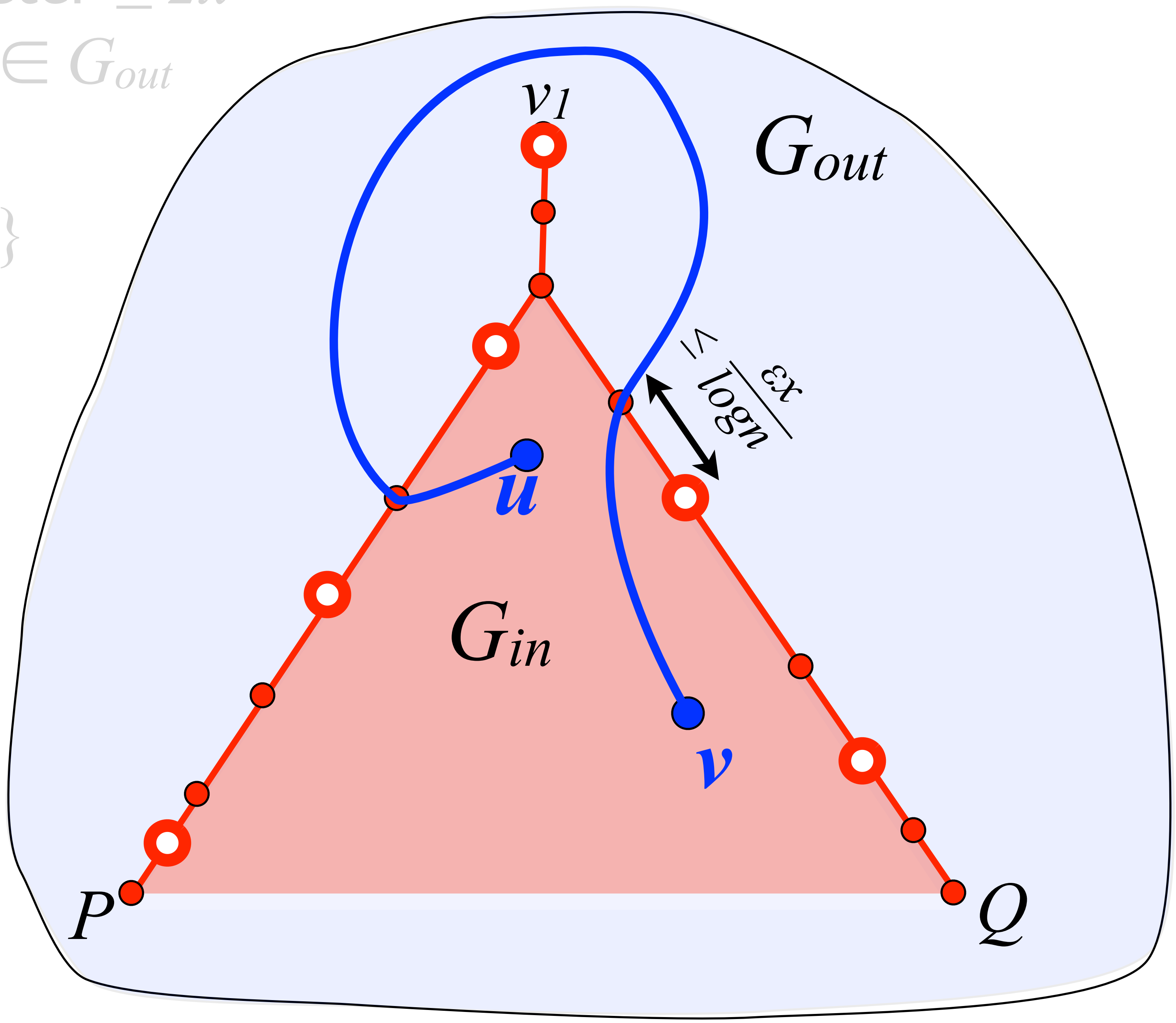
Recursive Algorithm

- Mark all nodes
- In $O(n)$ time find x s.t $x \leq \text{diameter} \leq 2x$

1. Find furthest pair $u \in G_{in}$ and $v \in G_{out}$
2. Find furthest pair in $G_{in} \setminus \{P,Q\}$
3. Find furthest pair in $G_{out} \setminus \{P,Q\}$

Choose $O((\log n) / \varepsilon)$

dense portals



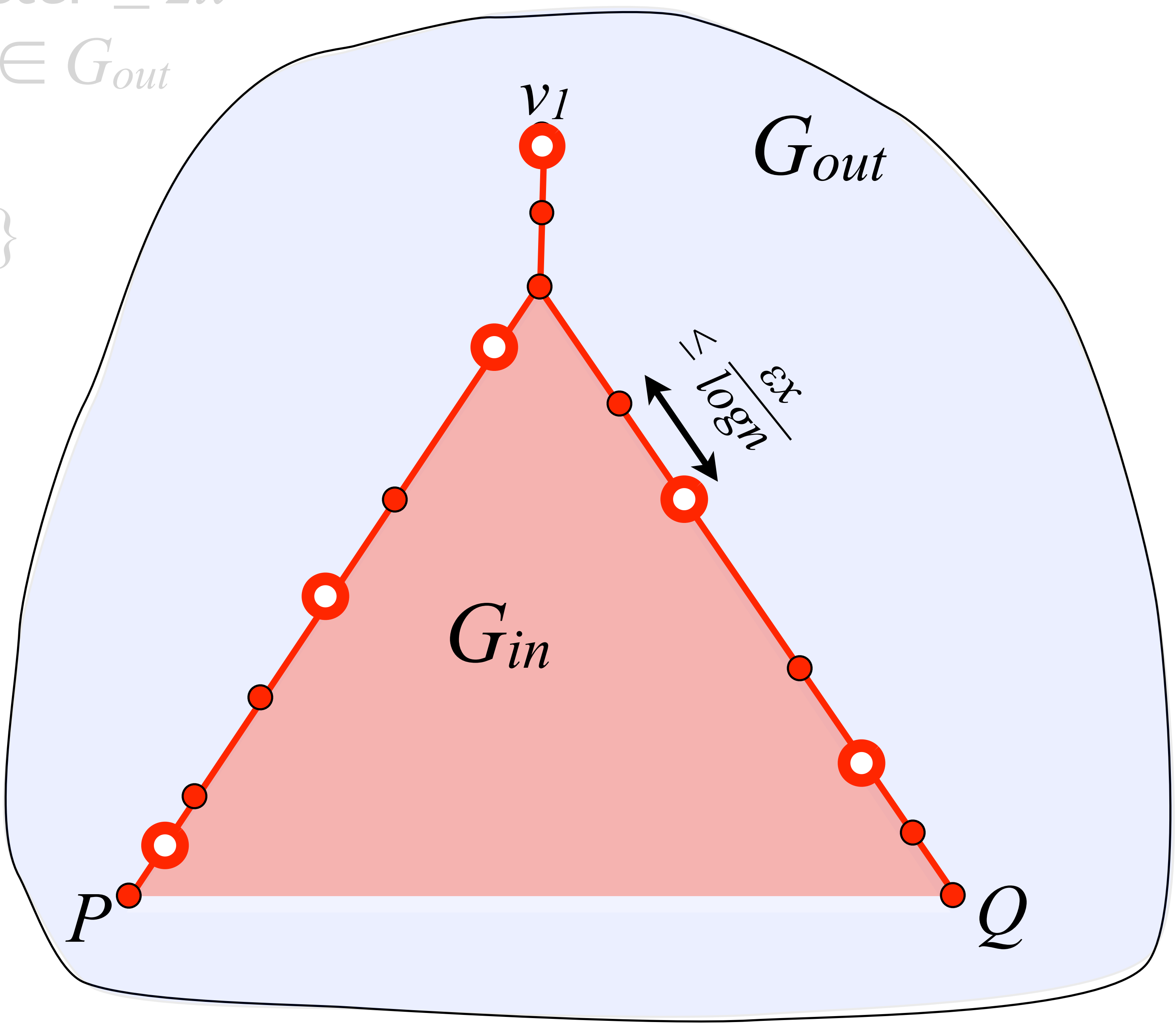
Recursive Algorithm

- Mark all nodes
- In $O(n)$ time find x s.t $x \leq$ diameter $\leq 2x$

1. Find furthest pair $u \in G_{in}$ and $v \in G_{out}$
2. Find furthest pair in $G_{in} \setminus \{P,Q\}$
3. Find furthest pair in $G_{out} \setminus \{P,Q\}$

Choose O((log n) / ε) *dense portals* ○

Compute all ○-to-○ shortest paths in G_{out}



Recursive Algorithm

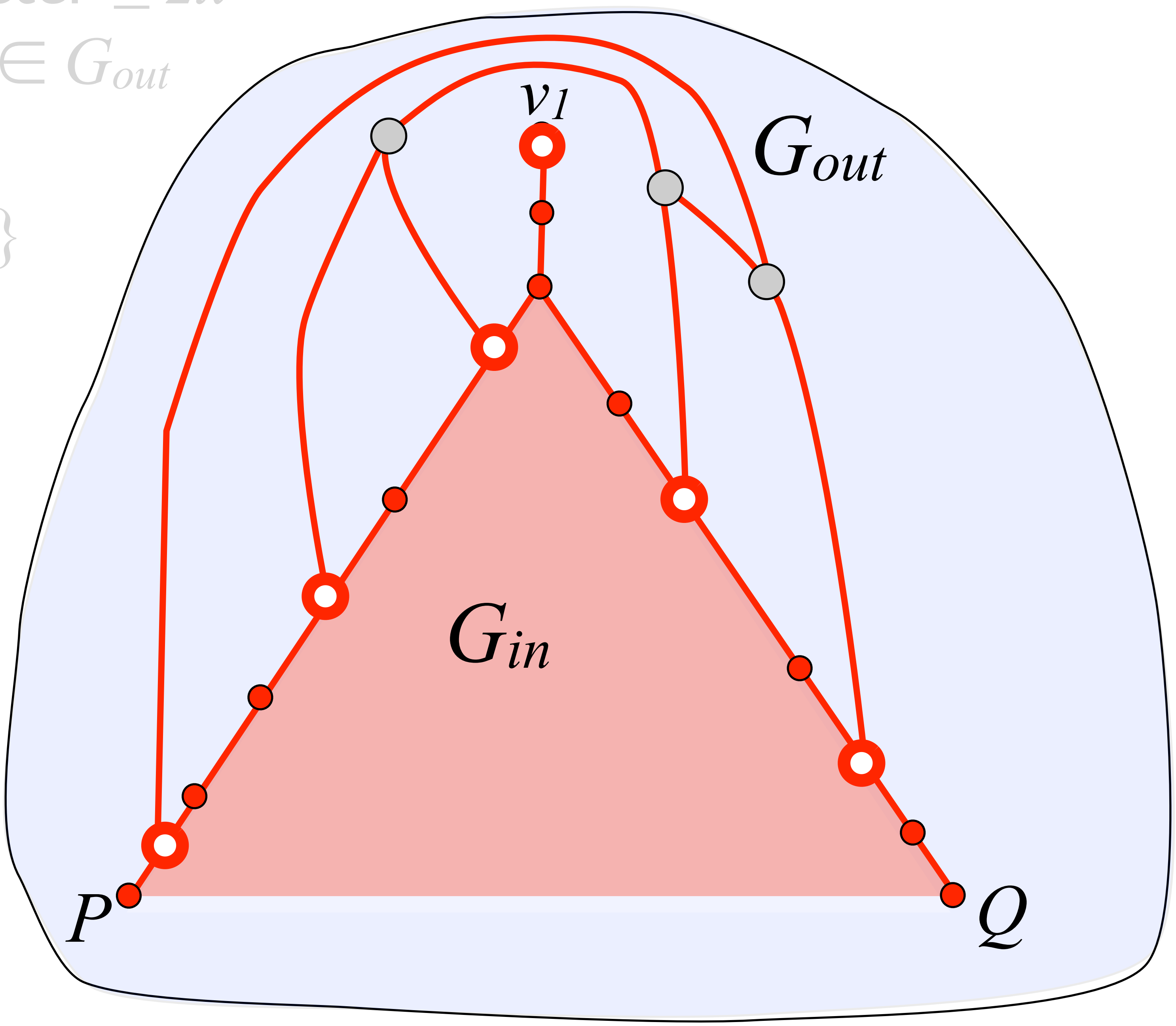
- Mark all nodes
- In $O(n)$ time find x s.t $x \leq$ diameter $\leq 2x$

1. Find furthest pair $u \in G_{in}$ and $v \in G_{out}$
2. Find furthest pair in $G_{in} \setminus \{P,Q\}$
3. Find furthest pair in $G_{out} \setminus \{P,Q\}$

Choose $O((\log n) / \varepsilon)$ *dense portals*

Compute all portal-to-portal shortest paths in G_{out}

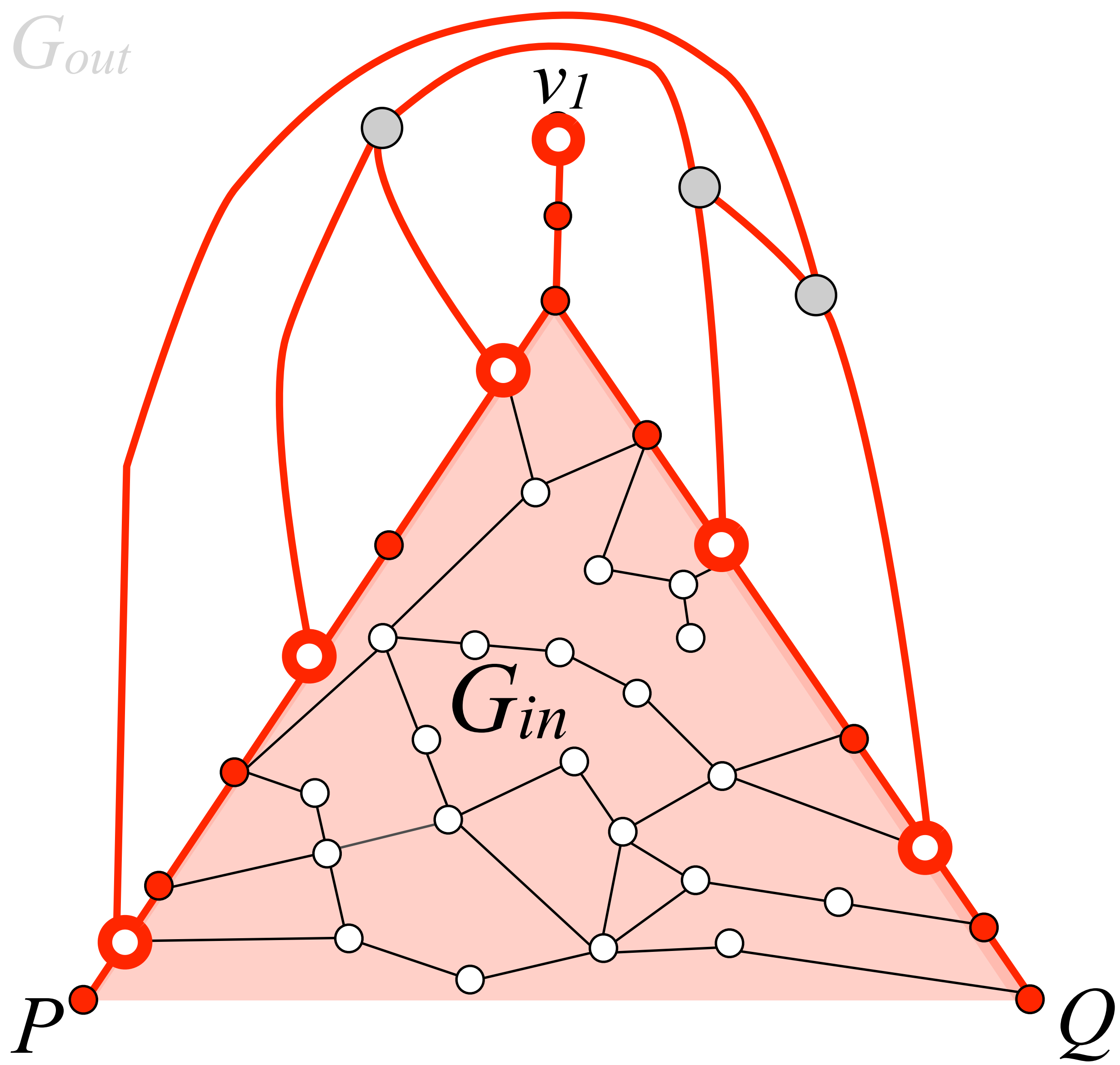
Contract degree-2 nodes
Unmark and append to G_{in}



Recursive Algorithm

- Mark all nodes
- In $O(n)$ time find x s.t $x \leq \text{diameter} \leq 2x$

1. Find furthest pair $u \in G_{in}$ and $v \in G_{out}$
2. Find furthest pair in $G_{in} \setminus \{P, Q\}$
3. Find furthest pair in $G_{out} \setminus \{P, Q\}$

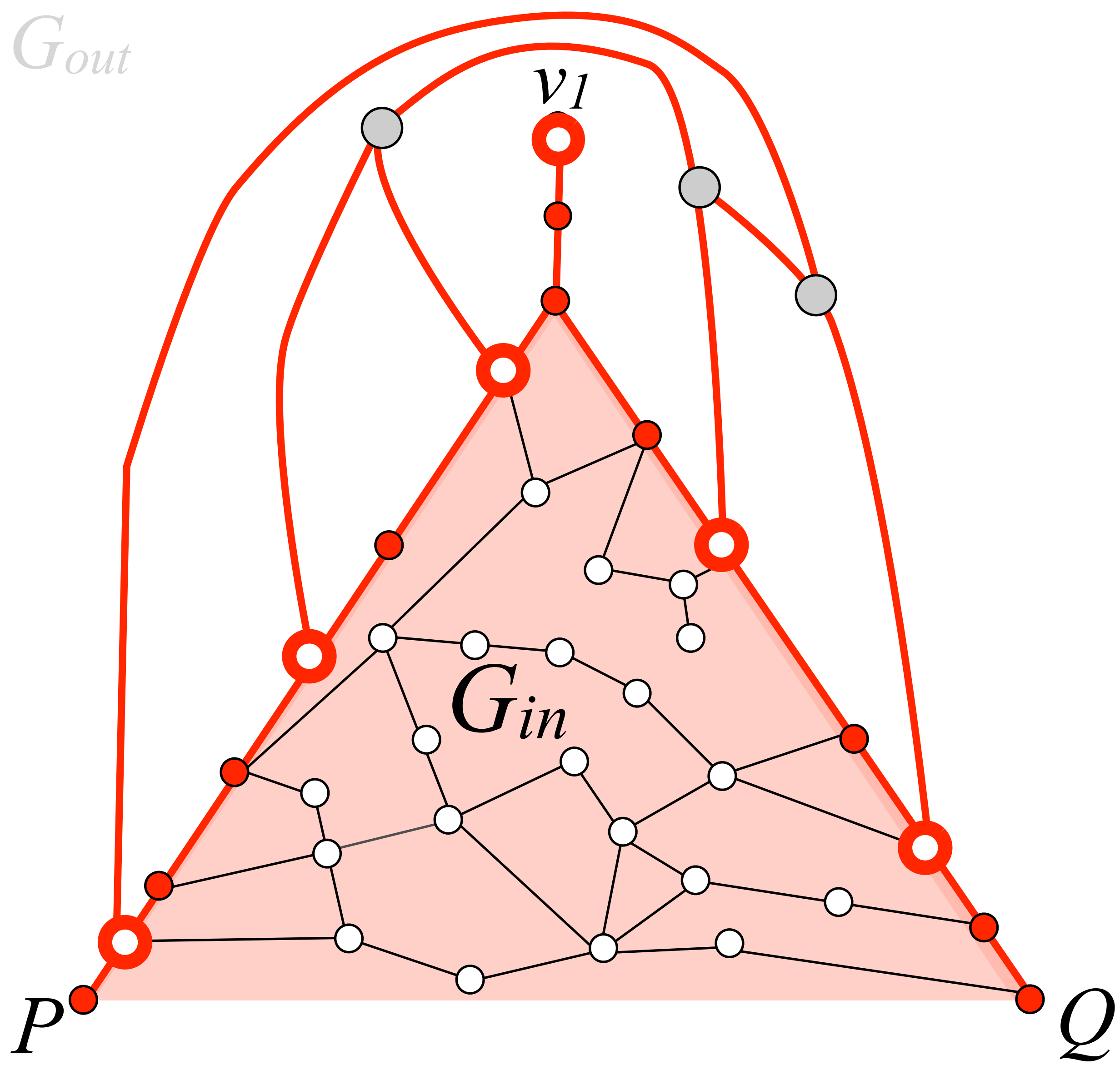


Recursive Algorithm

- Mark all nodes
- In $O(n)$ time find x s.t $x \leq$ diameter $\leq 2x$

1. Find furthest pair $u \in G_{in}$ and $v \in G_{out}$
2. Find furthest pair in $G_{in} \setminus \{P,Q\}$
3. Find furthest pair in $G_{out} \setminus \{P,Q\}$

This graph is still too big

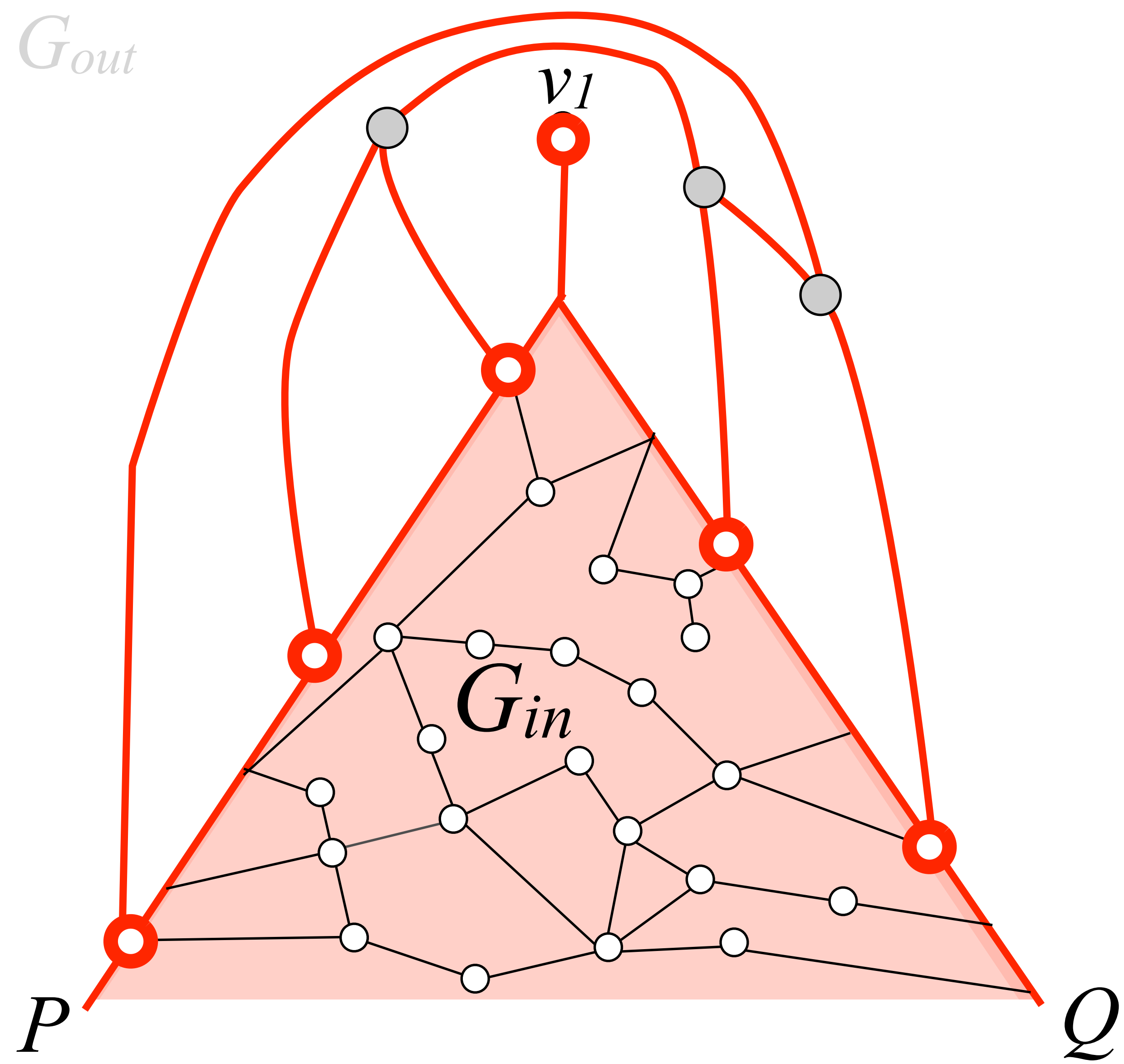


Recursive Algorithm

- Mark all nodes
- In $O(n)$ time find x s.t $x \leq \text{diameter} \leq 2x$

1. Find furthest pair $u \in G_{in}$ and $v \in G_{out}$
2. Find furthest pair in $G_{in} \setminus \{P,Q\}$
3. Find furthest pair in $G_{out} \setminus \{P,Q\}$

This graph is still too big

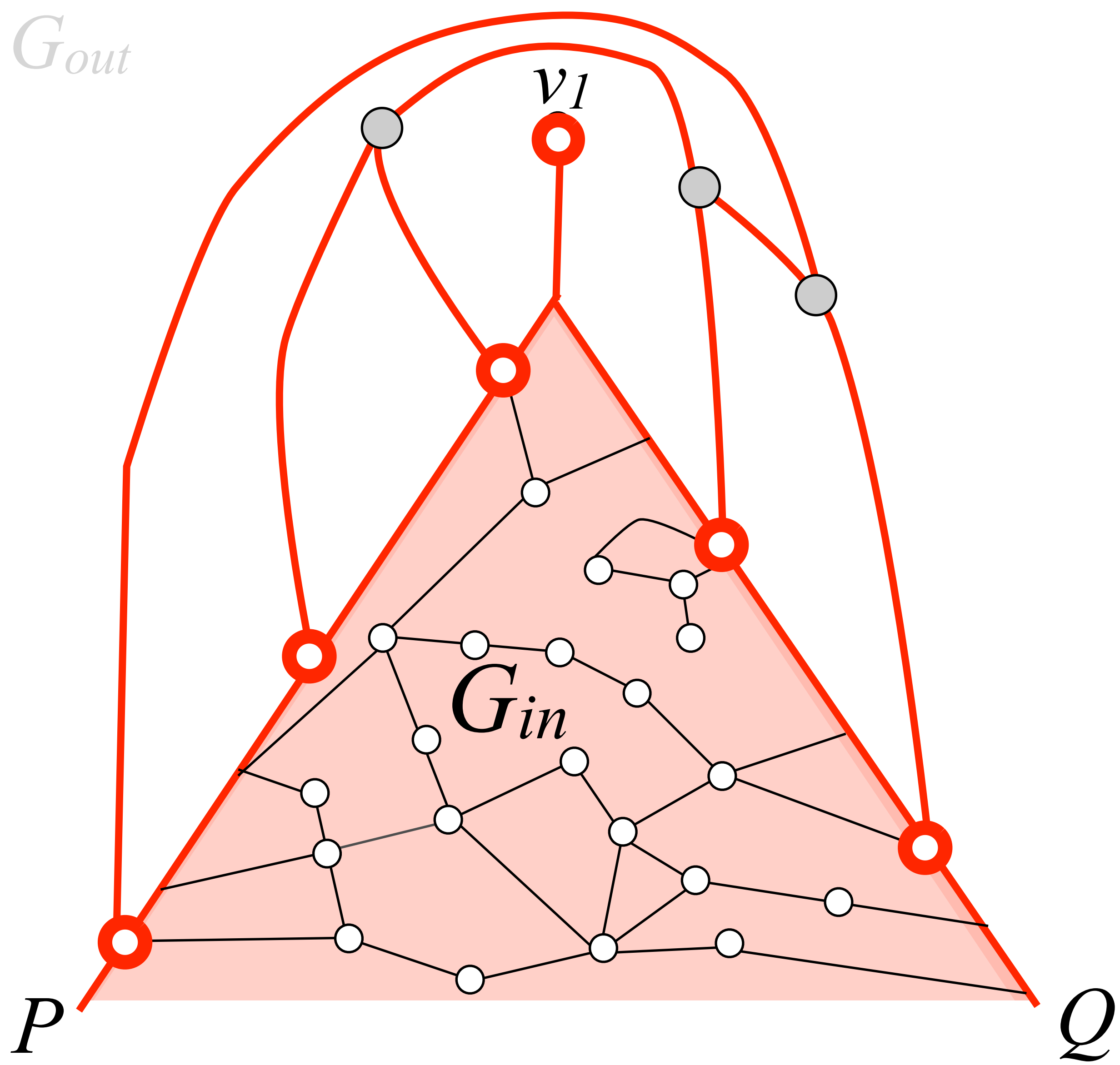


Recursive Algorithm

- Mark all nodes
- In $O(n)$ time find x s.t $x \leq \text{diameter} \leq 2x$

1. Find furthest pair $u \in G_{in}$ and $v \in G_{out}$
2. Find furthest pair in $G_{in} \setminus \{P,Q\}$
3. Find furthest pair in $G_{out} \setminus \{P,Q\}$

This graph is still too big

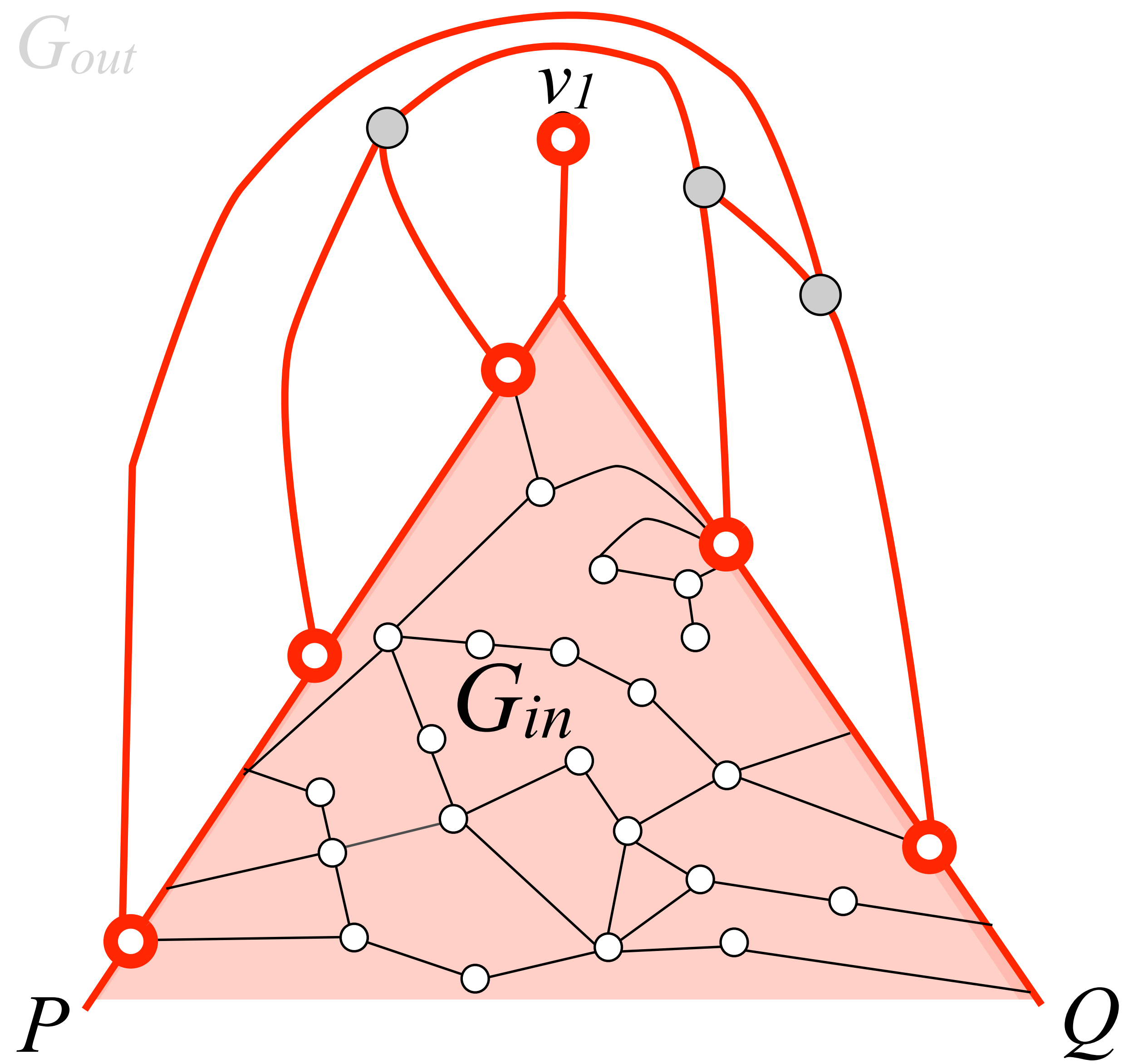


Recursive Algorithm

- Mark all nodes
- In $O(n)$ time find x s.t $x \leq \text{diameter} \leq 2x$

1. Find furthest pair $u \in G_{in}$ and $v \in G_{out}$
2. Find furthest pair in $G_{in} \setminus \{P,Q\}$
3. Find furthest pair in $G_{out} \setminus \{P,Q\}$

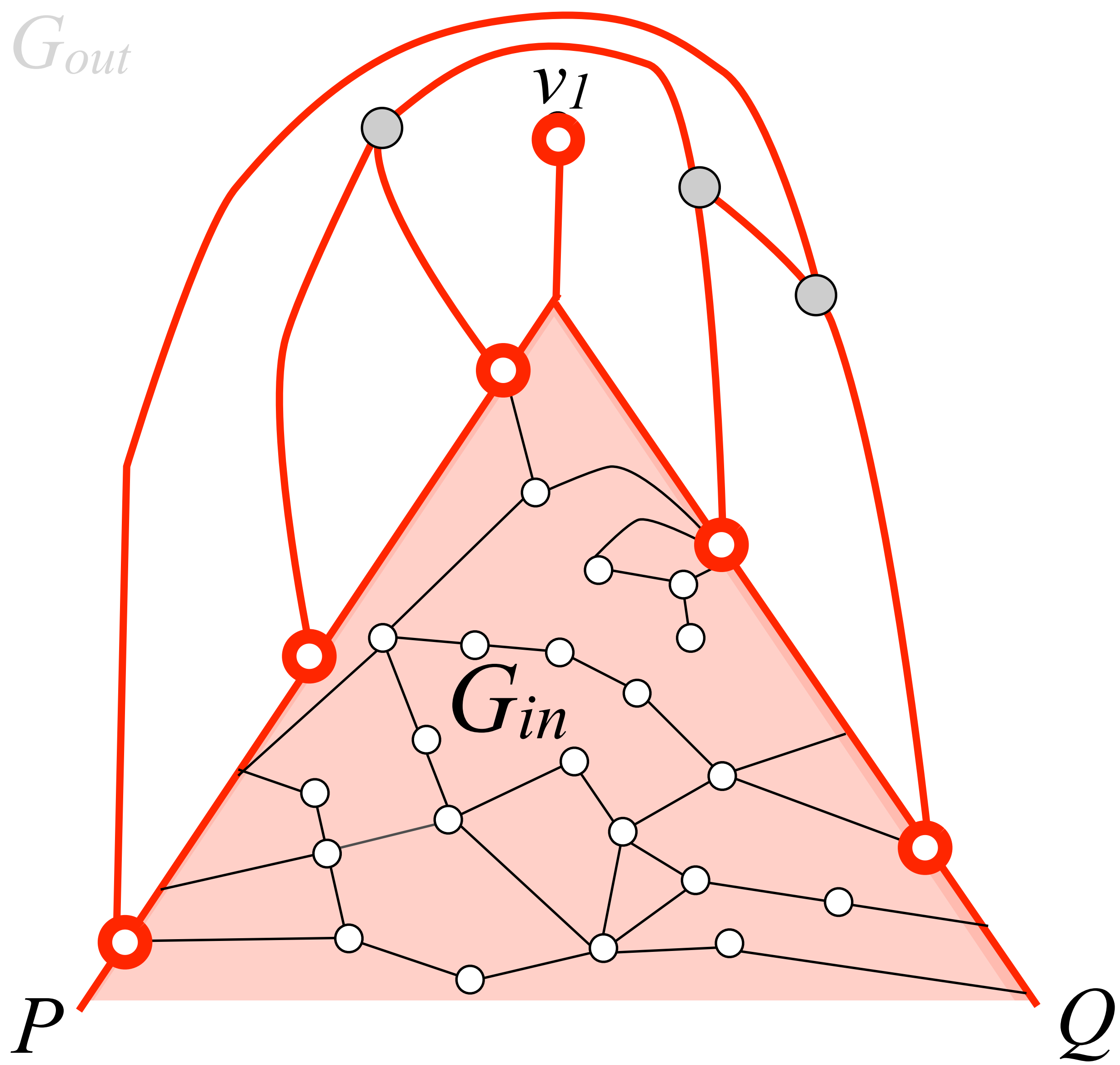
This graph is still too big



Recursive Algorithm

- Mark all nodes
- In $O(n)$ time find x s.t $x \leq \text{diameter} \leq 2x$

1. Find furthest pair $u \in G_{in}$ and $v \in G_{out}$
2. Find furthest pair in $G_{in} \setminus \{P,Q\}$
3. Find furthest pair in $G_{out} \setminus \{P,Q\}$



Open Problem

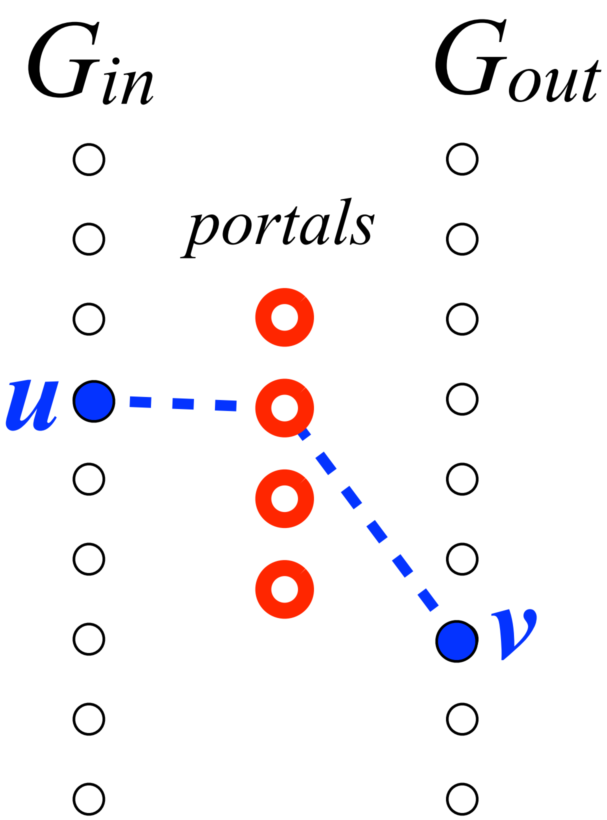
- We obtained $\tilde{O}(f(\varepsilon)\cdot n)$ time where $f(\varepsilon) = 2^{O(1/\varepsilon)}$

Open Problem

- We obtained $\tilde{O}(f(\varepsilon)\cdot n)$ time where $f(\varepsilon) = 2^{O(1/\varepsilon)}$
  - $f(\varepsilon)$ = poly$(1/\varepsilon)$ say $f(\varepsilon) = (1/\varepsilon)^c$ would immediately imply an <u>exact</u> algorithm for diameter in $O(n^{2-\varepsilon})$ when diameter is bounded by $n^{1/c}$

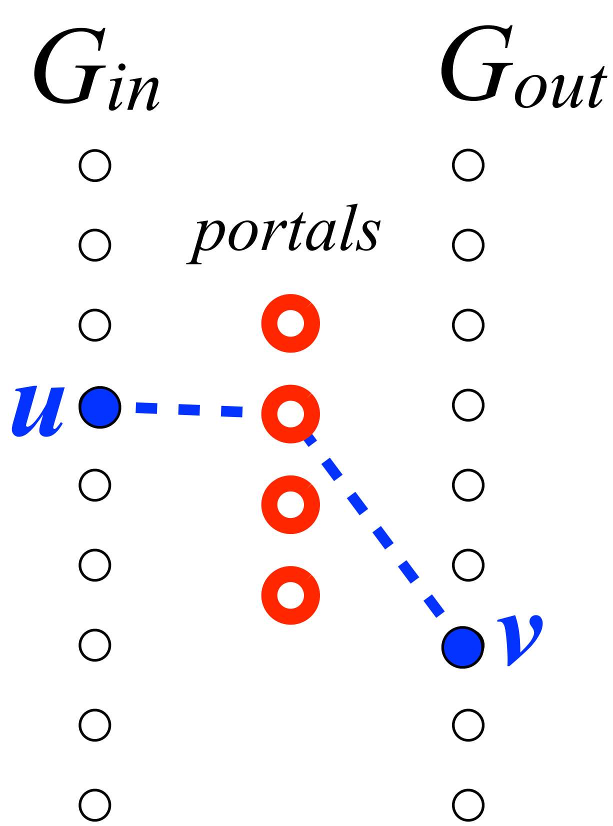
Open Problem

- We obtained $\tilde{O}(f(\varepsilon)\cdot n)$ time where $f(\varepsilon) = 2^{O(1/\varepsilon)}$
  - $f(\varepsilon)$ = poly$(1/\varepsilon)$ say $f(\varepsilon) = (1/\varepsilon)^c$ would immediately imply an <u>exact</u> algorithm for diameter in $O(n^{2-\varepsilon})$ when diameter is bounded by $n^{1/c}$



Open Problem

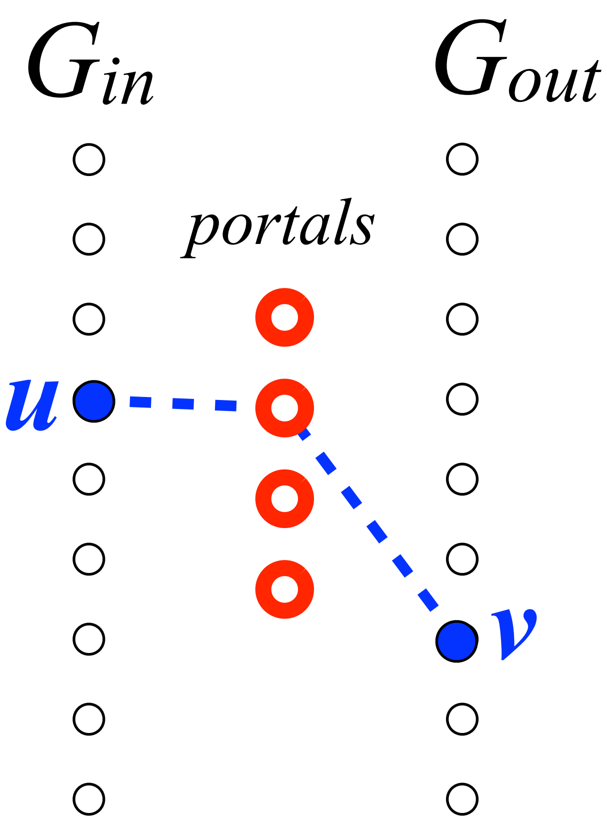
- We obtained $\tilde{O}(f(\varepsilon)\cdot n)$ time where $f(\varepsilon) = 2^{O(1/\varepsilon)}$
  - $f(\varepsilon)$ = poly($1/\varepsilon$) say $f(\varepsilon) = (1/\varepsilon)^c$ would immediately imply an exact algorithm for diameter in $O(n^{2-\varepsilon})$ when diameter is bounded by $n^{1/c}$



General $n \times 1/\varepsilon \times n$ tripartite graph requires $O(n\cdot\log^{1/\varepsilon} n)$ [Cabello, Knauer 2009], we get $O(n\cdot 2^{1/\varepsilon})$

Open Problem

- We obtained $\tilde{O}(f(\varepsilon)\cdot n)$ time where $f(\varepsilon) = 2^{O(1/\varepsilon)}$
  - $f(\varepsilon) = \text{poly}(1/\varepsilon)$ say $f(\varepsilon) = (1/\varepsilon)^c$ would immediately imply an <u>exact</u> algorithm for diameter in $O(n^{2-\varepsilon})$ when diameter is bounded by $n^{1/c}$

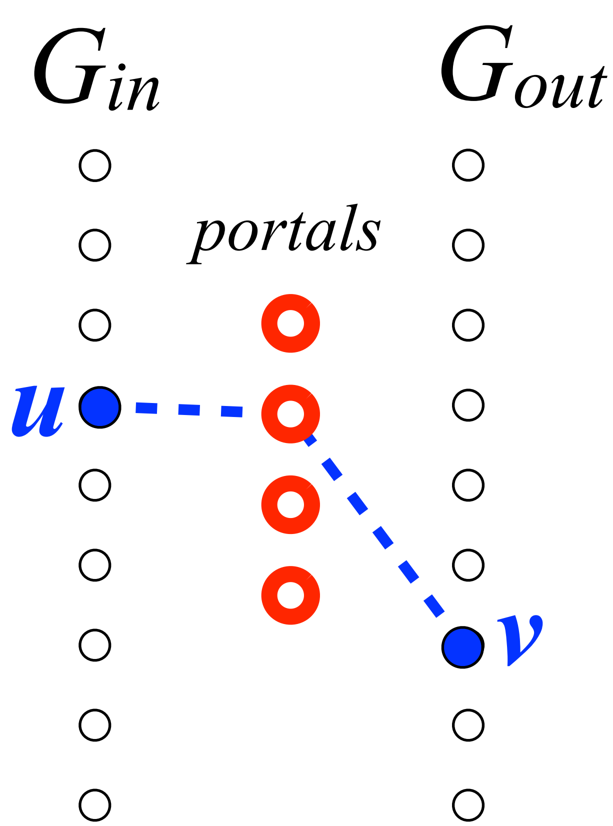


General $n \times 1/\varepsilon \times n$ tripartite graph requires $O(n \cdot \log^{1/\varepsilon} n)$ [Cabello, Knauer 2009], we get $O(n \cdot 2^{1/\varepsilon})$

bounded lengths

Open Problem

- We obtained $\tilde{O}(f(\varepsilon)\cdot n)$ time where $f(\varepsilon) = 2^{O(1/\varepsilon)}$
  - $f(\varepsilon)$ = poly$(1/\varepsilon)$ say $f(\varepsilon) = (1/\varepsilon)^c$ would immediately imply an <u>exact</u> algorithm for diameter in $O(n^{2-\varepsilon})$ when diameter is bounded by $n^{1/c}$



General $n \times 1/\varepsilon \times n$ tripartite graph requires $O(n\cdot\log^{1/\varepsilon} n)$ [Cabello, Knauer 2009], we get $O(n\cdot 2^{1/\varepsilon})$

bounded lengths

(1) We can settle for an approximation
(2) Lengths correspond to planar distances (Monge)
(3) Range max can be easier than sum

Thank You!