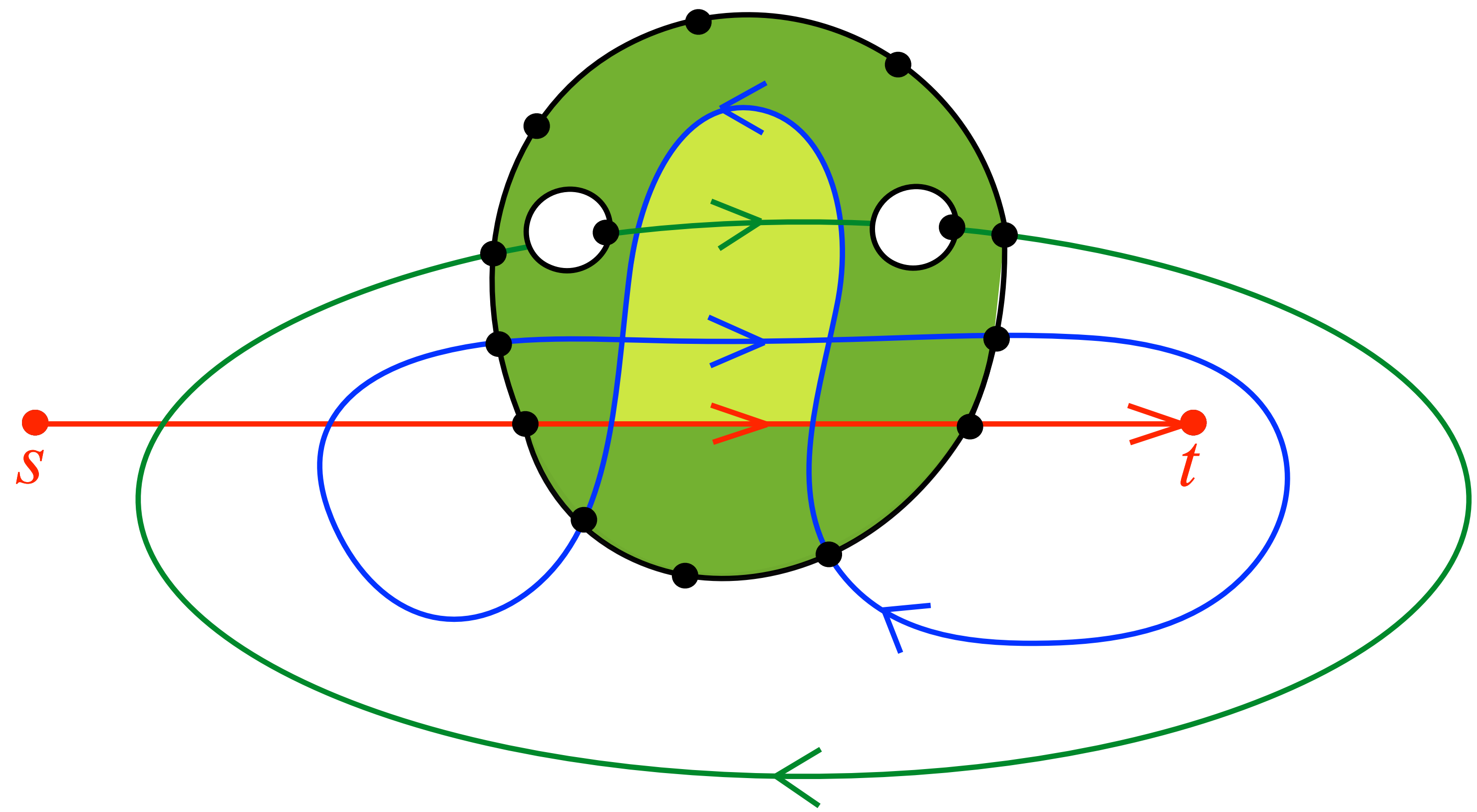
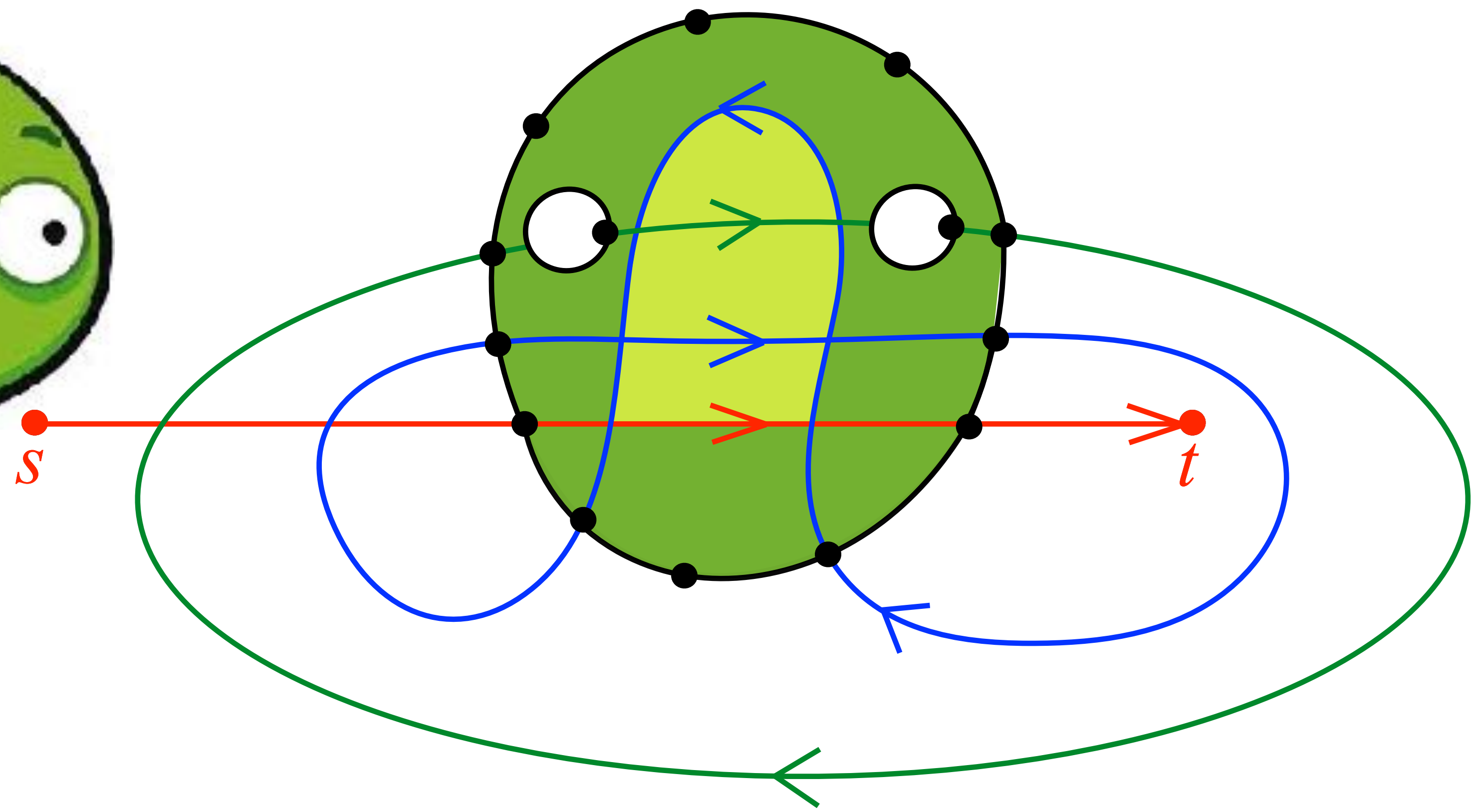
Minimum Cut of Directed Planar Graphs in O(nloglogn) Time



Shay Mozes, Kirill Nikolaev, Yahav Nussbaum, Oren Weimann

Minimum Cut of Directed Planar Graphs in O(nloglogn) Time



Shay Mozes, Kirill Nikolaev, Yahav Nussbaum, Oren Weimann

Minimum Cut and Minimum st-cut in Planar Graphs

- Undirected min st-cut: $O(n \log\log n)$
  [Italiano, Nussbaum, Sankowski, Wulff-Nilsen STOC 2011]

- Undirected min cut: $O(n \log\log n)$
  [Łącki, Sankowski ESA 2011]

- Directed min st-cut: $O(n \log n)$
  [Borradaile, Klein SODA 2006]

- Directed min cut $O(n \log^2 n)$
  [Chalermsook, Fakcharoenphol, Nanongkai SODA 2004]

Minimum Cut and Minimum st-cut in Planar Graphs

- Undirected min st-cut: $O(n \log\log n)$
  [Italiano, Nussbaum, Sankowski, Wulff-Nilsen STOC 2011]

- Undirected min cut: $O(n \log\log n)$
  [Łącki, Sankowski ESA 2011]

- Directed min st-cut: $O(n \log n)$
  [Borradaile, Klein SODA 2006]

- Directed min cut $O(n \log^2 n)$ $O(n \log\log n)$
  [Chalermsook, Fakcharoenphol, Nanongkai SODA 2004]

Minimum Cut and Minimum st-cut in Planar Graphs

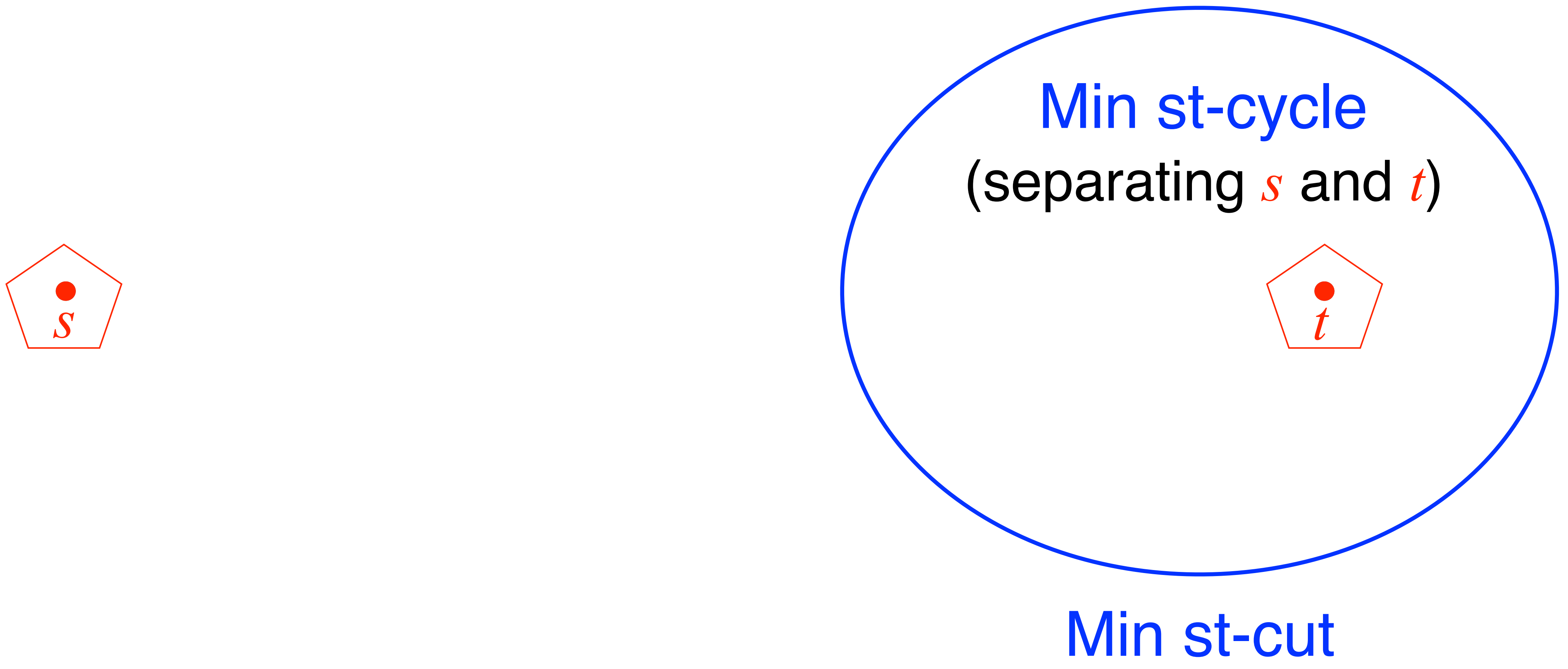
- Undirected min st-cut: $O(n \log\log n)$
  [Italiano, Nussbaum, Sankowski, Wulff-Nilsen STOC 2011]
- Undirected min cut: $O(n \log\log n)$
  [Łącki, Sankowski ESA 2011]
- Directed min cut $O(n \log^2 n)$ $O(n \log\log n)$
  [Chalermsook, Fakcharoenphol, Nanongkai SODA 2004]
- Directed min st-cut: $O(n \log n)$
  [Borradaile, Klein SODA 2006]

Cuts and Cycles in Planar Graphs

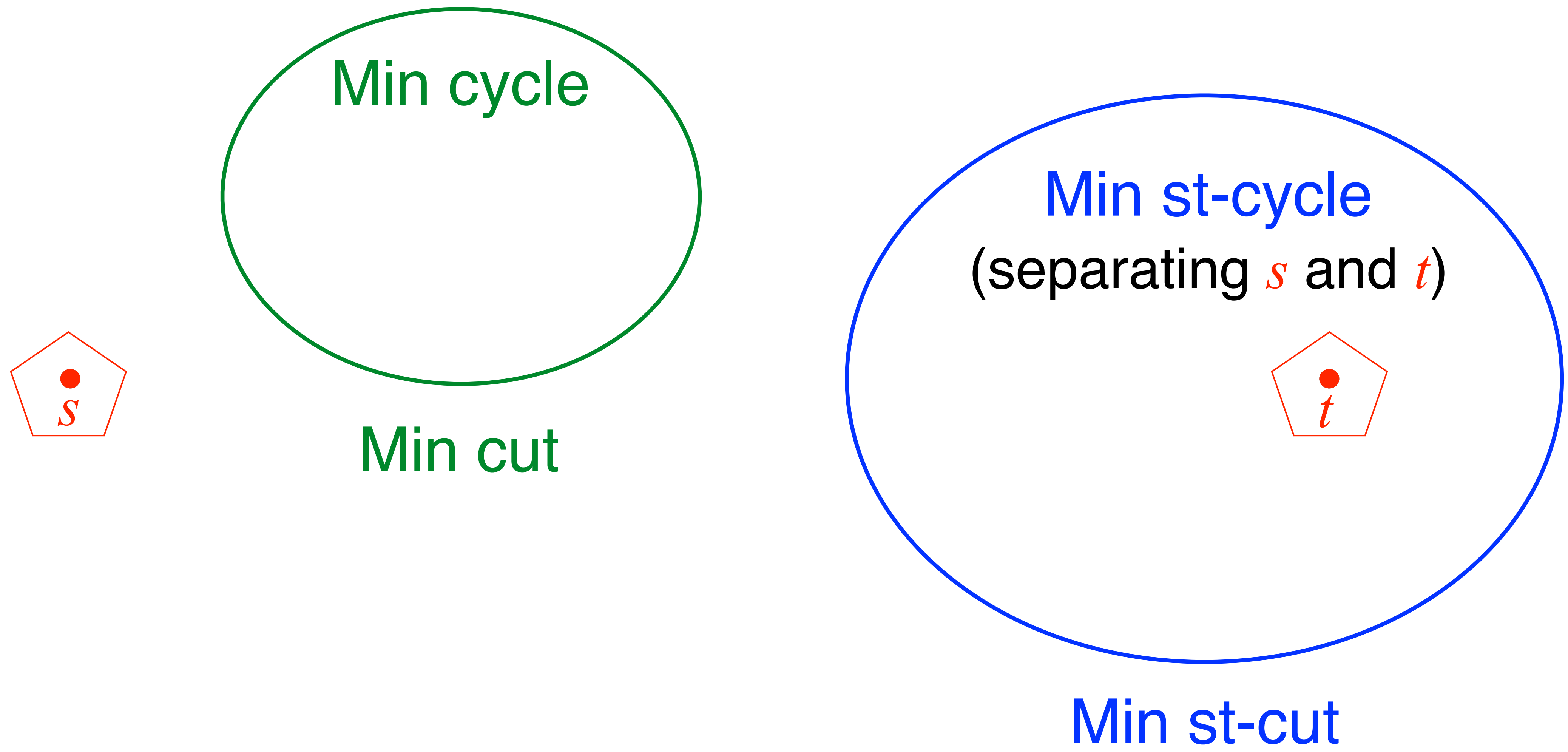
Cuts and Cycles in Planar Graphs

Cuts and Cycles in Planar Graphs

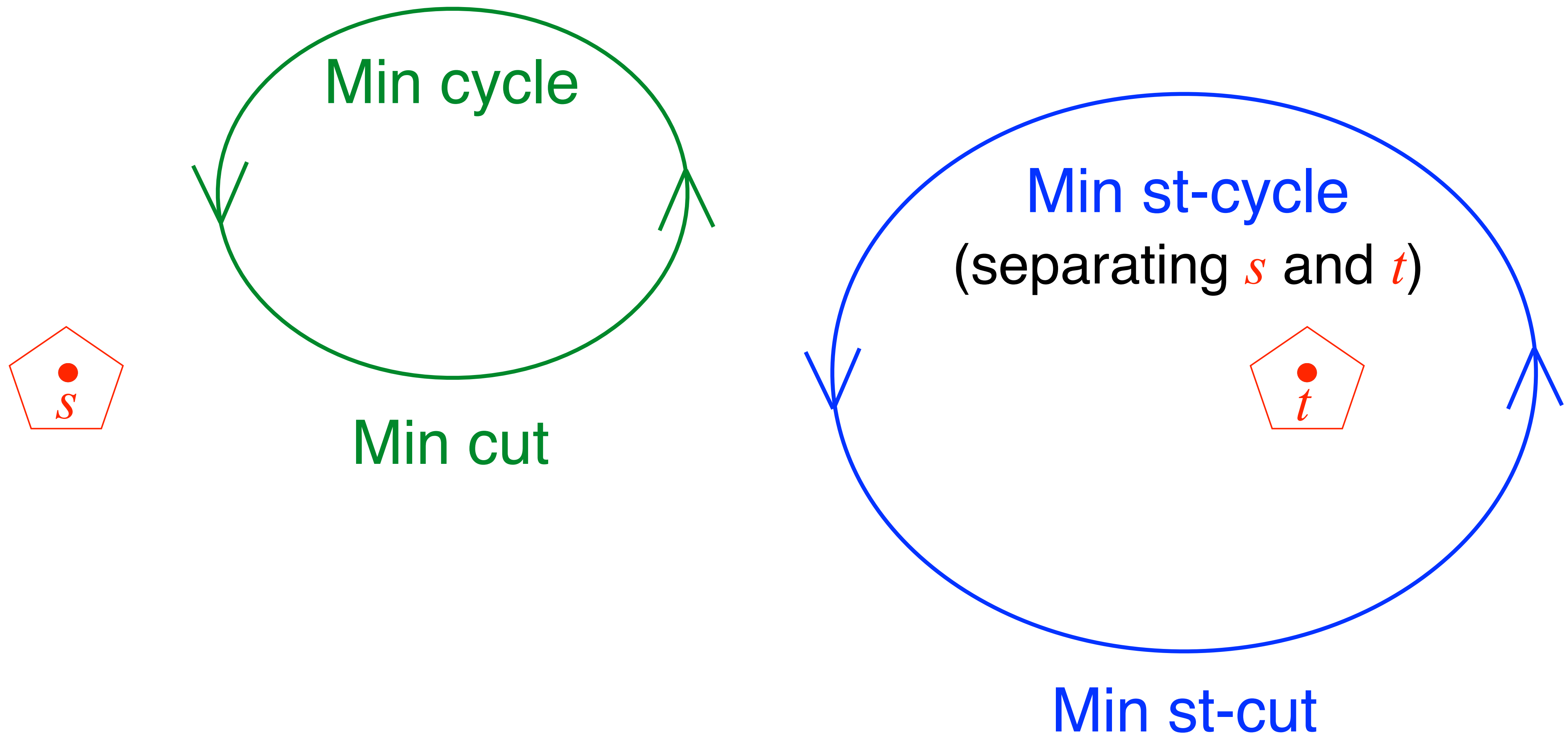
Cuts and Cycles in Planar Graphs



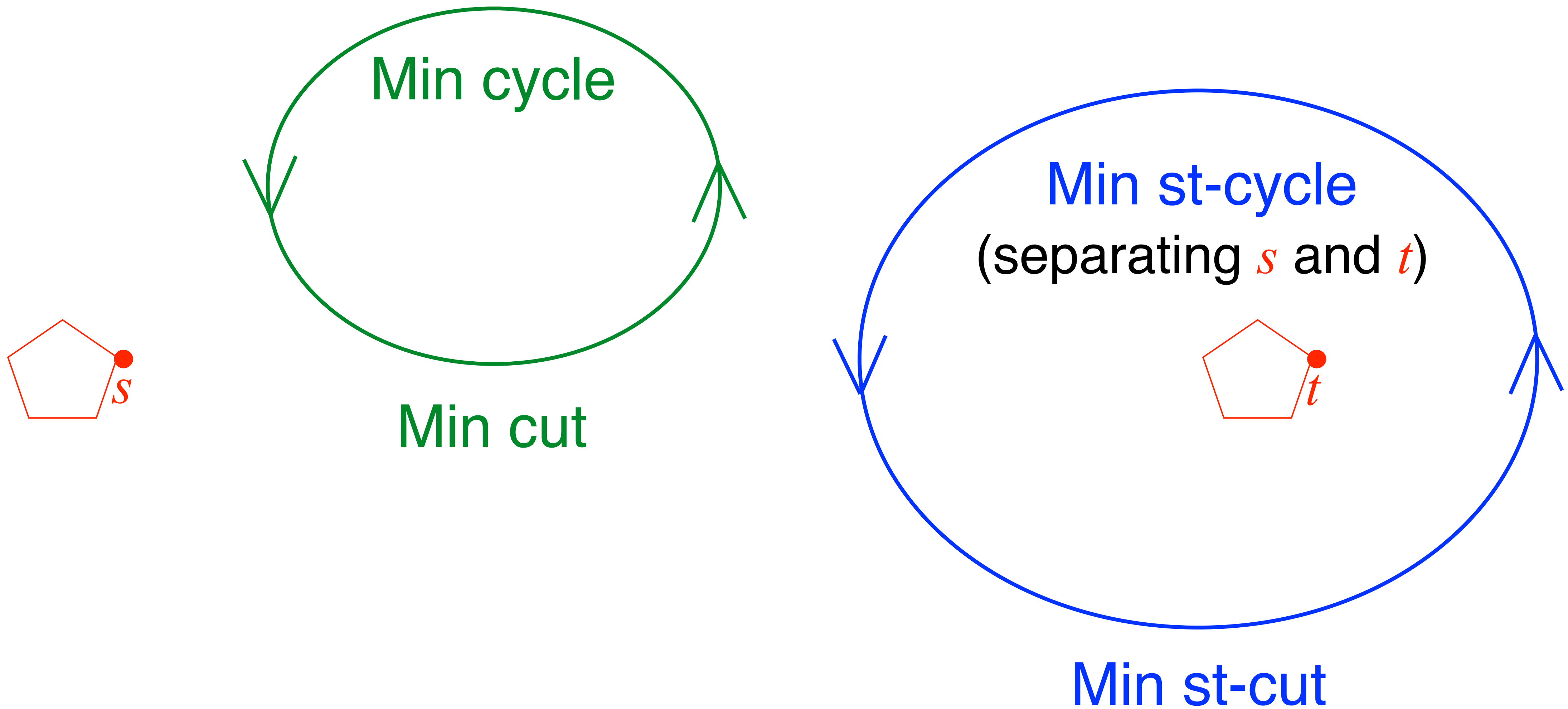
Cuts and Cycles in Planar Graphs



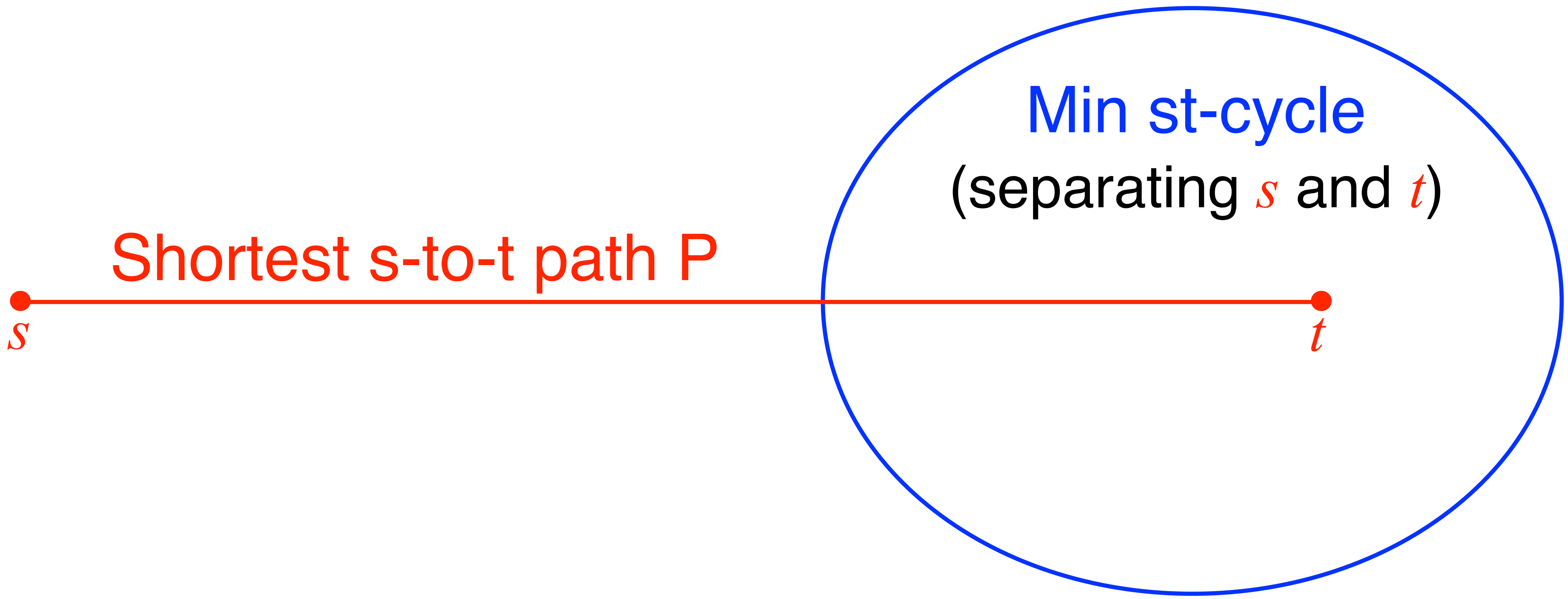
Cuts and Cycles in Planar Graphs



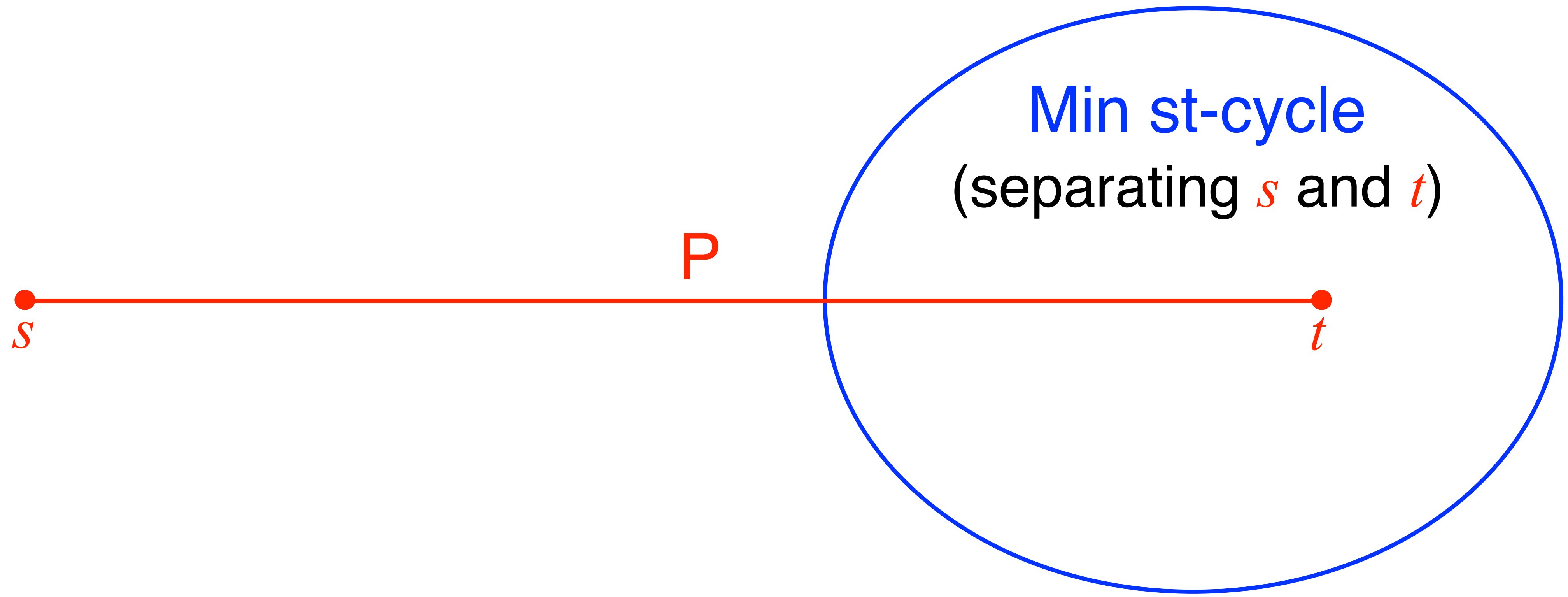
Cuts and Cycles in Planar Graphs



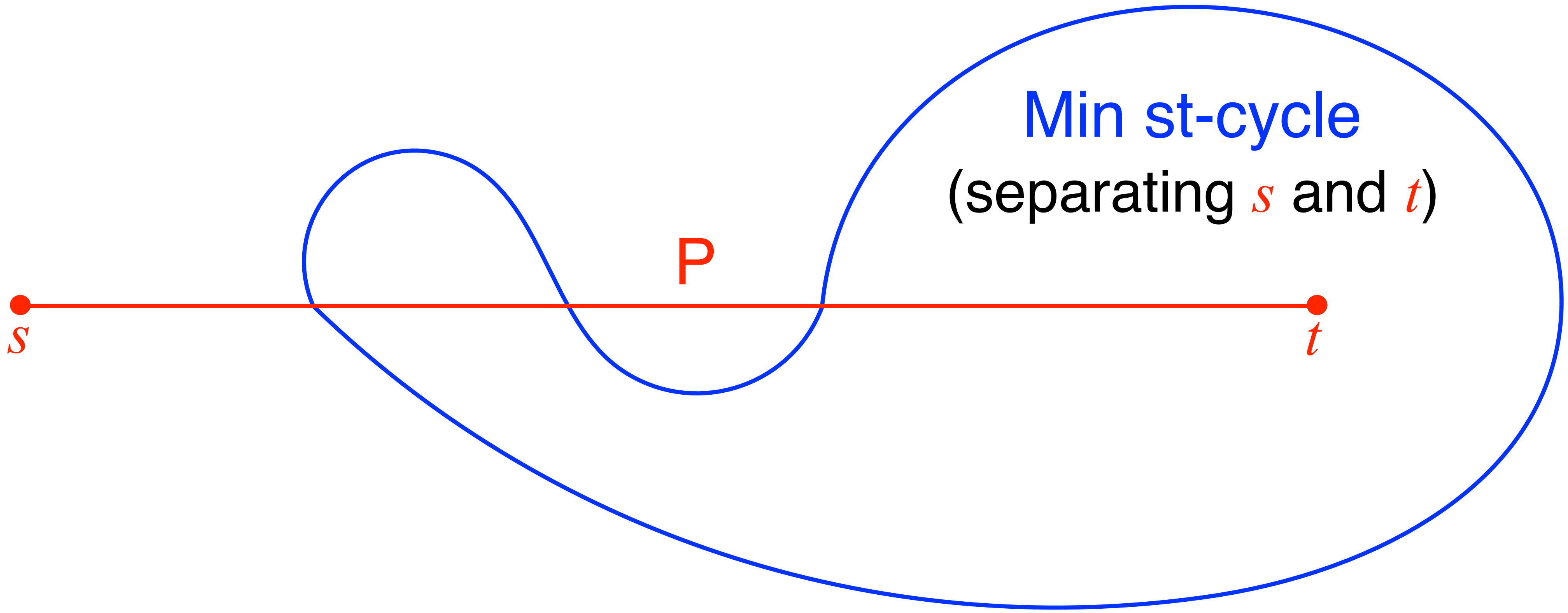
Cuts and Cycles in Planar Graphs



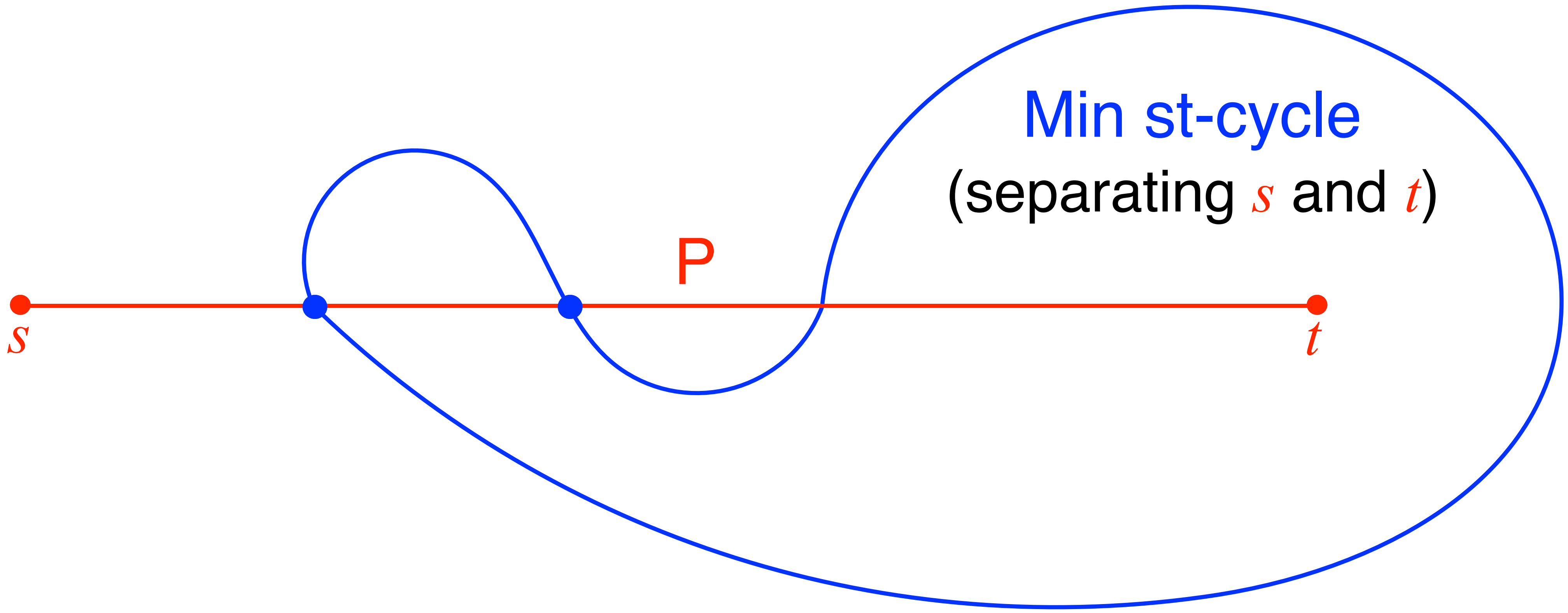
Undirected Min st-cycle Crosses P Once



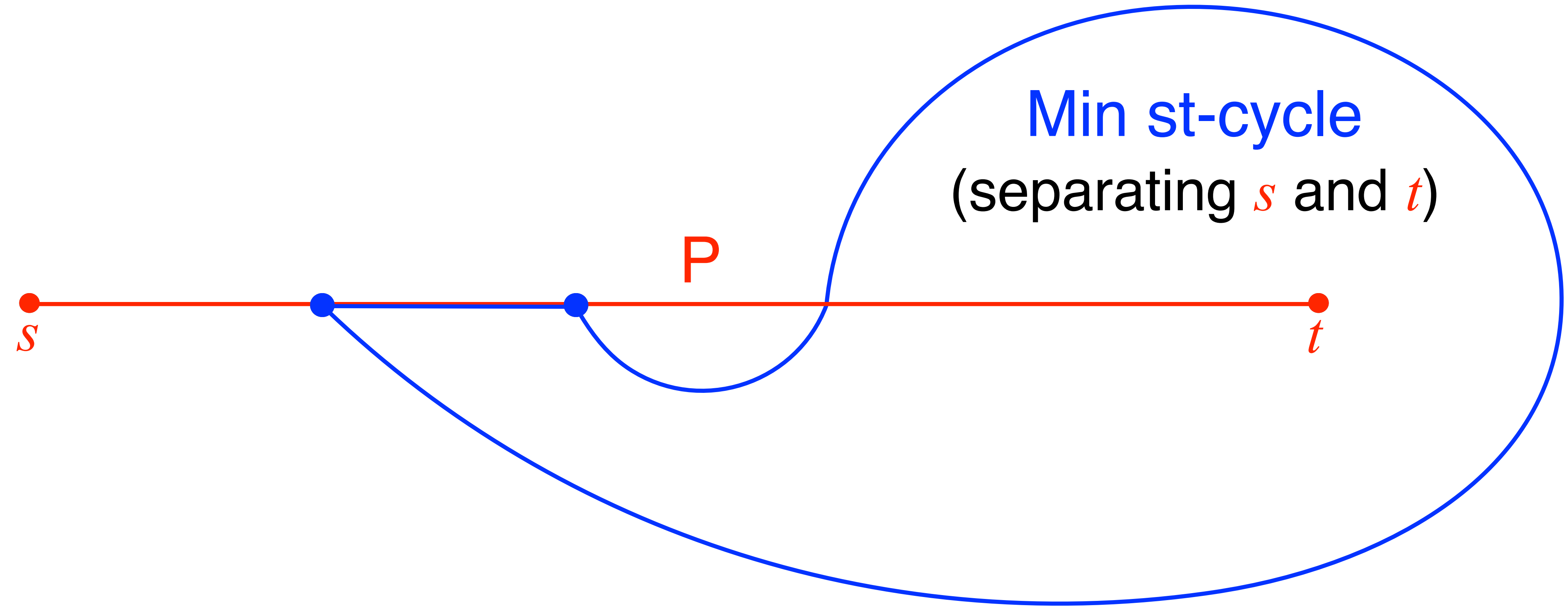
Undirected Min st-cycle Crosses P Once



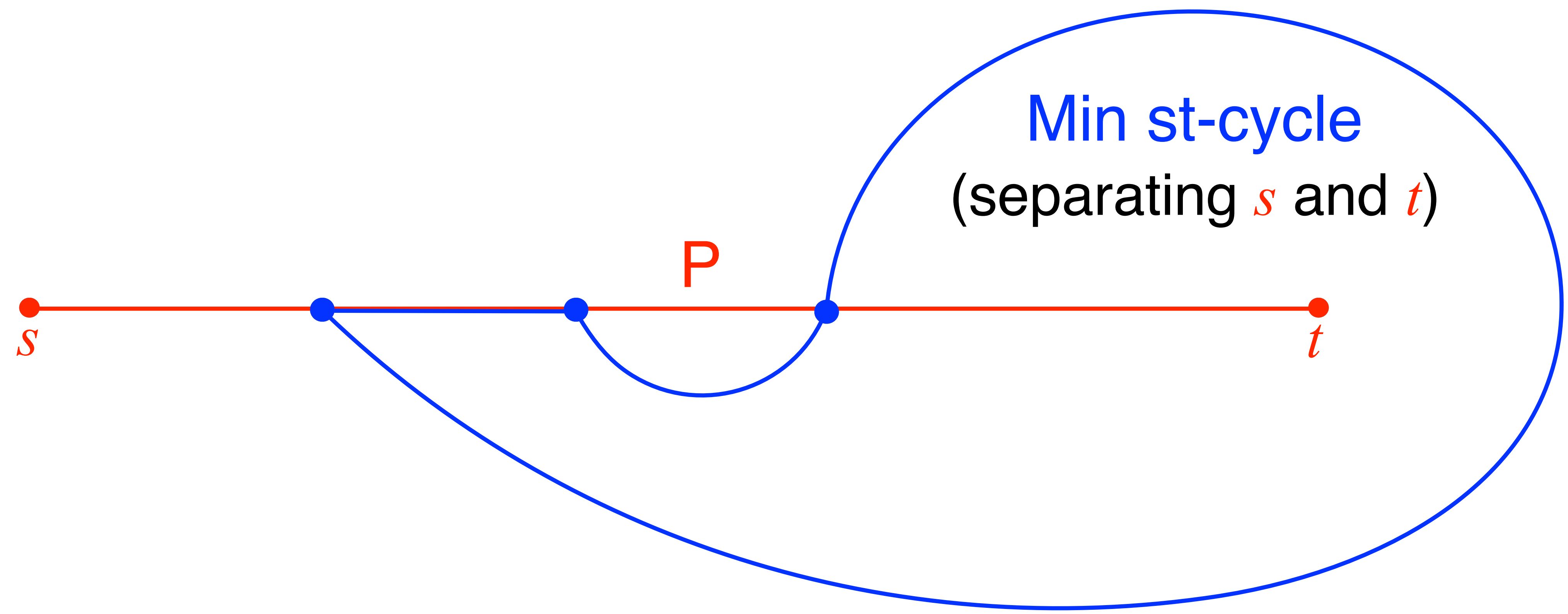
Undirected Min st-cycle Crosses P Once



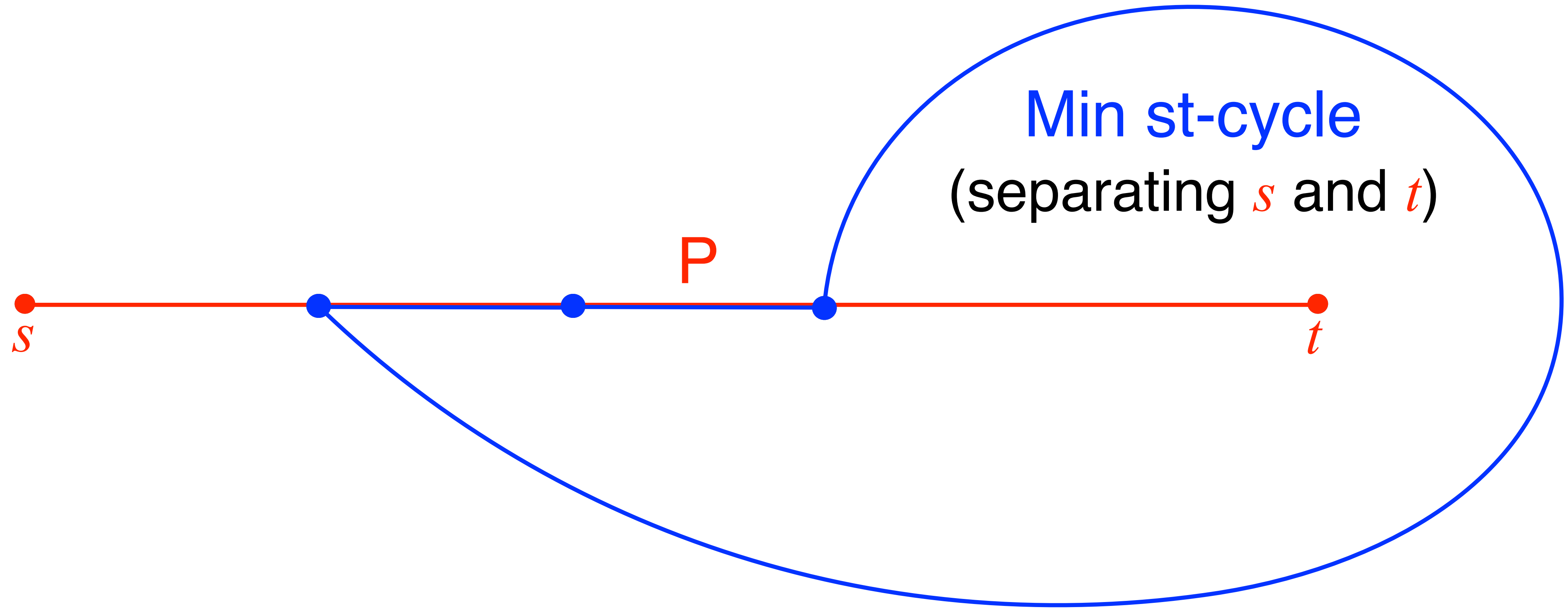
Undirected Min st-cycle Crosses P Once



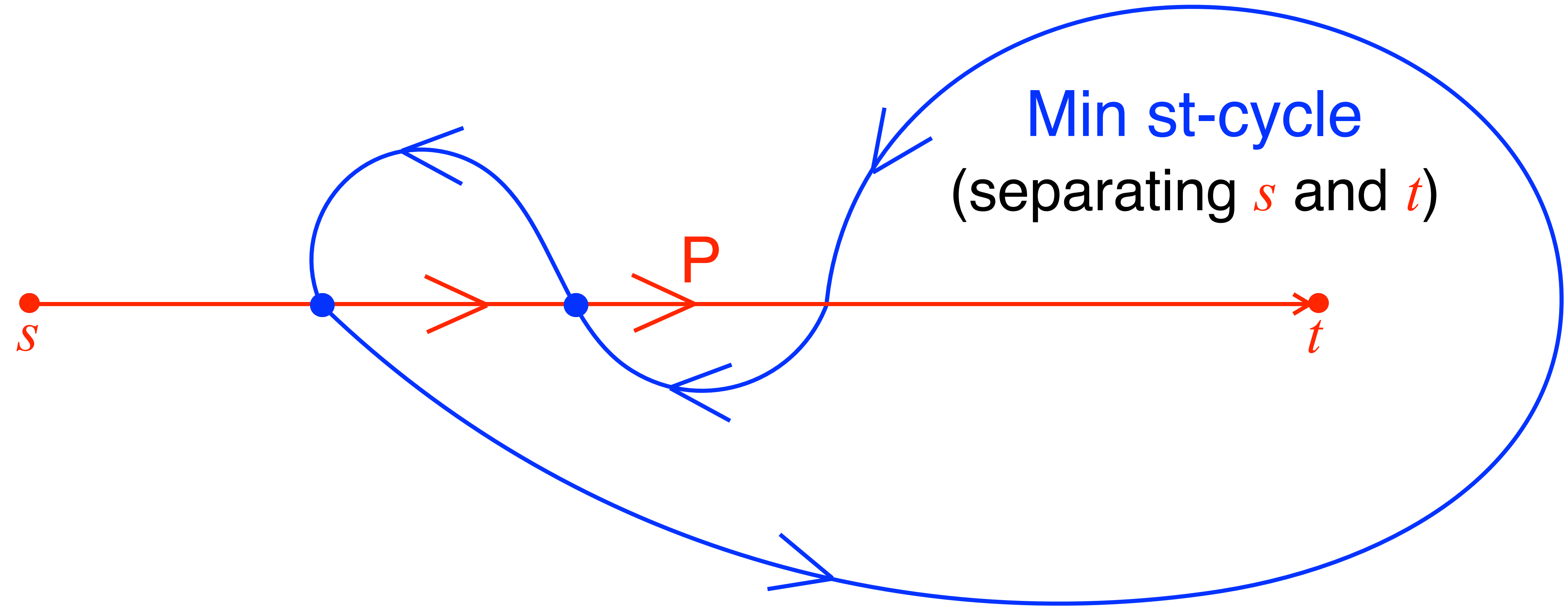
Undirected Min st-cycle Crosses P Once



Undirected Min st-cycle Crosses P Once

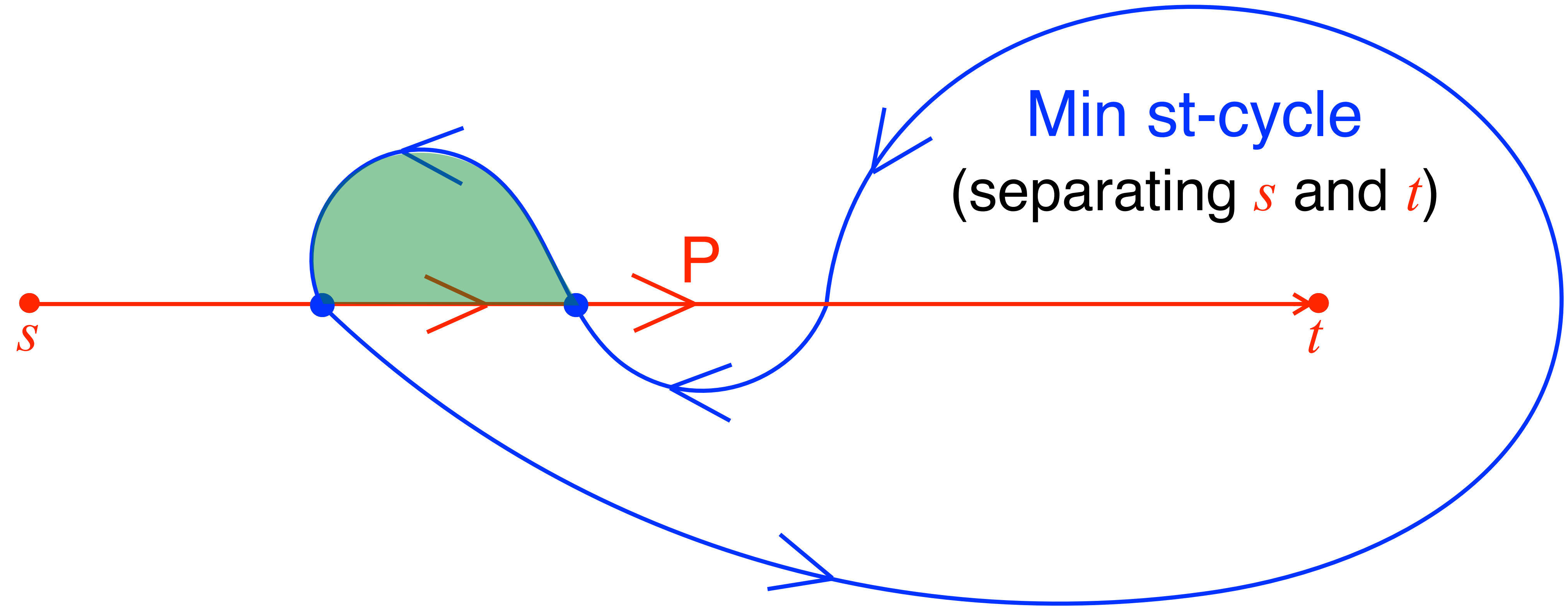


Directed Min st-cycle Crosses P Multiple Times



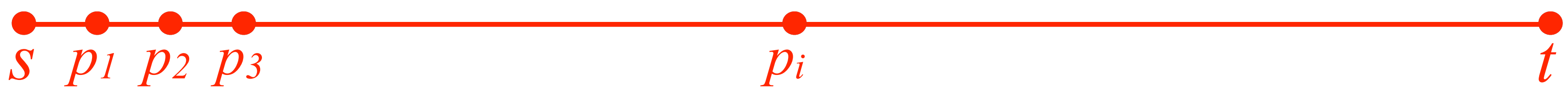
Directed Min st-cycle Crosses P Multiple Times

Directed Min cycle Crosses P at most Once!

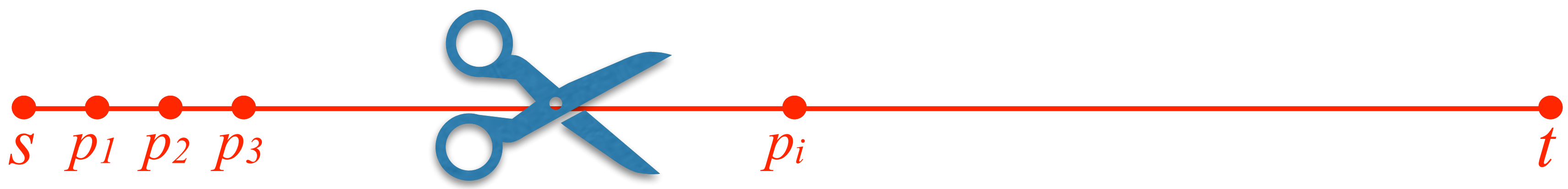


Undirected: Reif's Algorithm [Reif 1983]

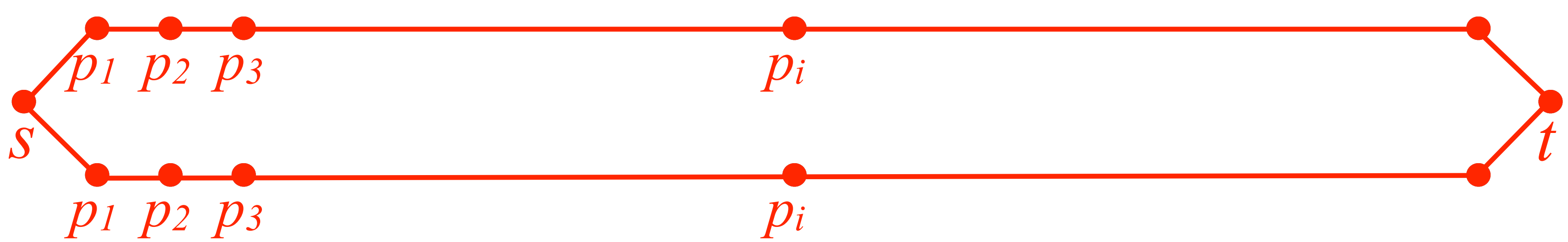
Undirected: Reif's Algorithm [Reif 1983]



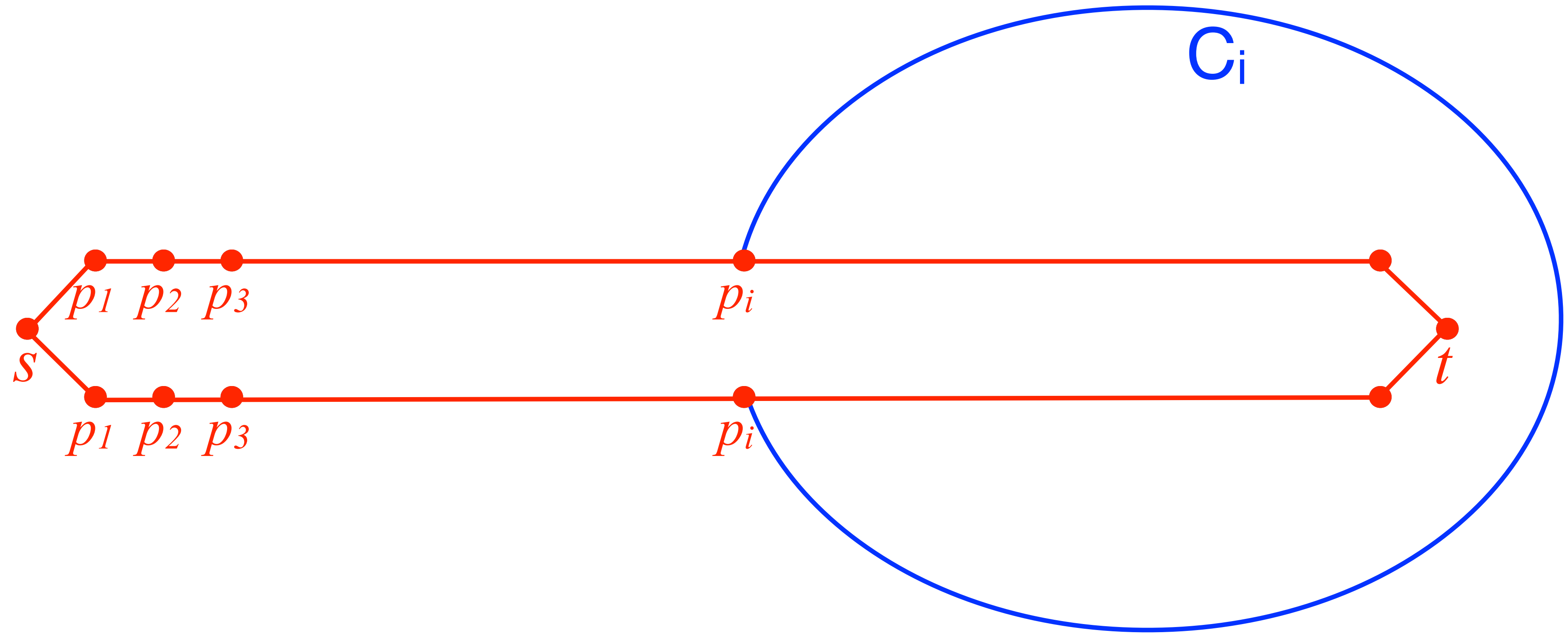
Undirected: Reif's Algorithm [Reif 1983]



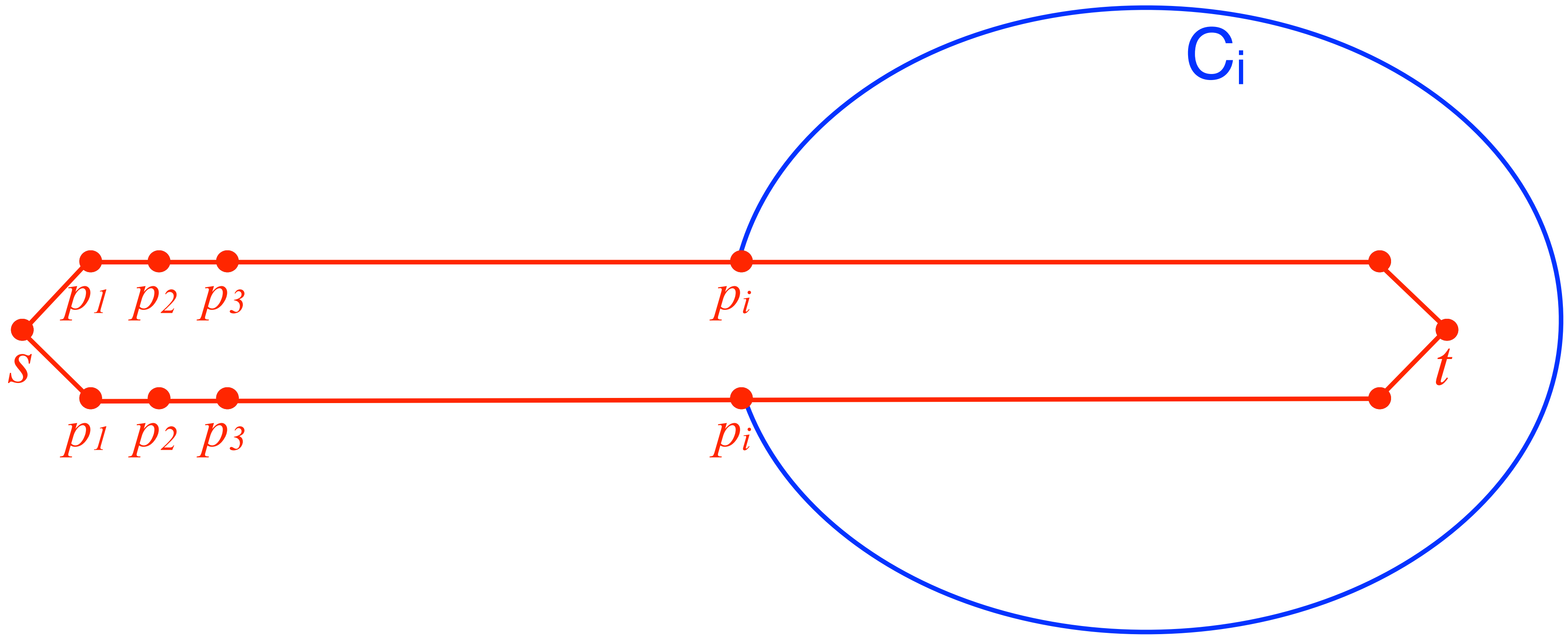
Undirected: Reif's Algorithm [Reif 1983]



Undirected: Reif's Algorithm [Reif 1983]

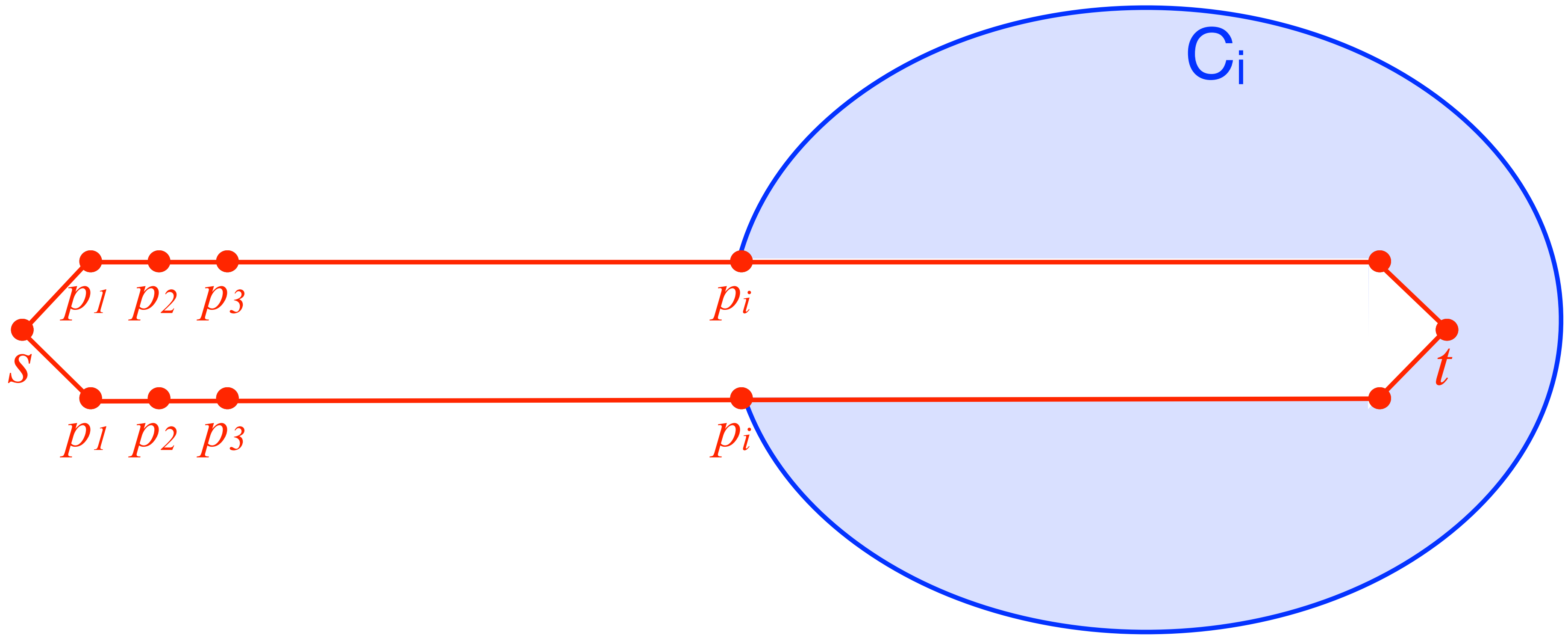


Undirected: Reif's Algorithm [Reif 1983]



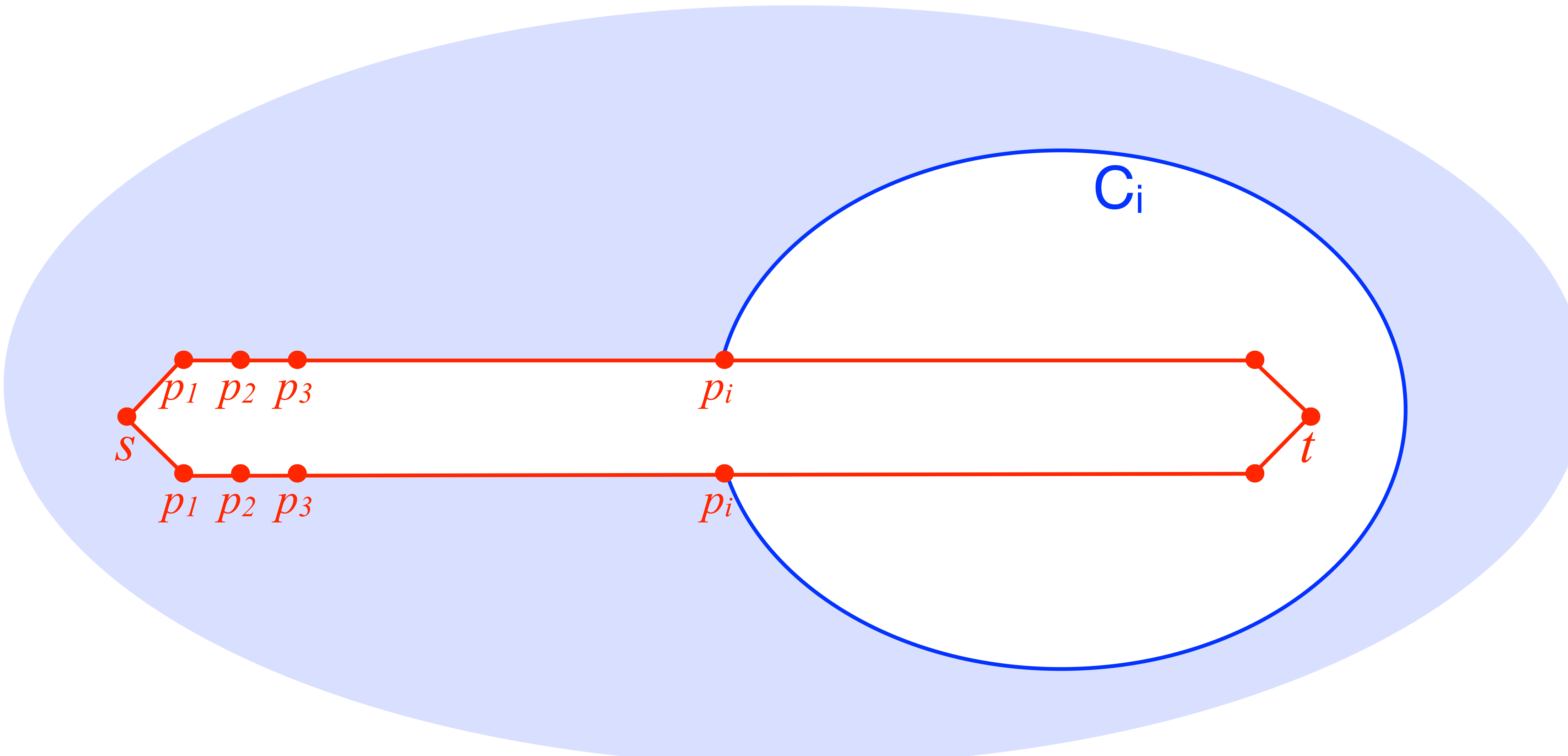
SSSP computation in O(n) time [Henzinger et al 1997]

Undirected: Reif's Algorithm [Reif 1983]



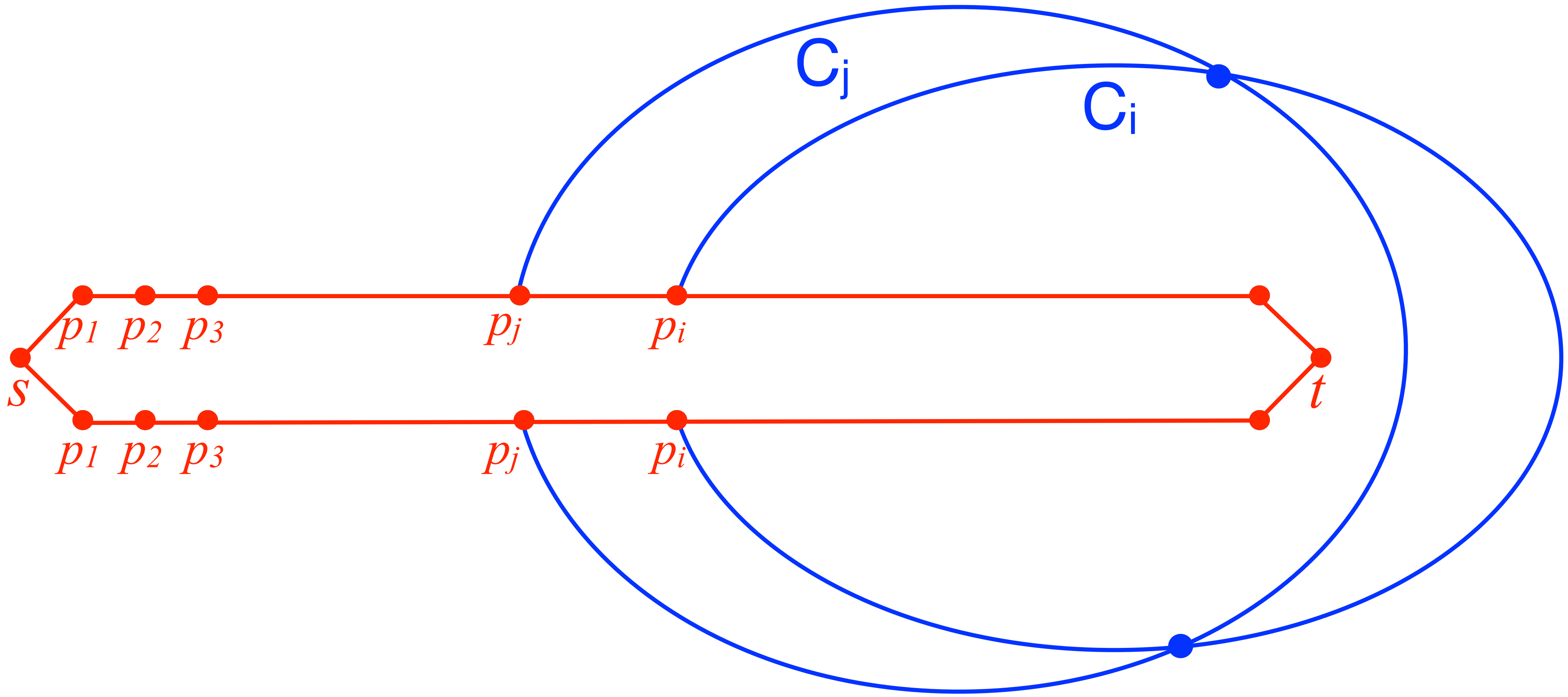
SSSP computation in O(n) time [Henzinger et al 1997]

Undirected: Reif's Algorithm [Reif 1983]



SSSP computation in O(n) time [Henzinger et al 1997]

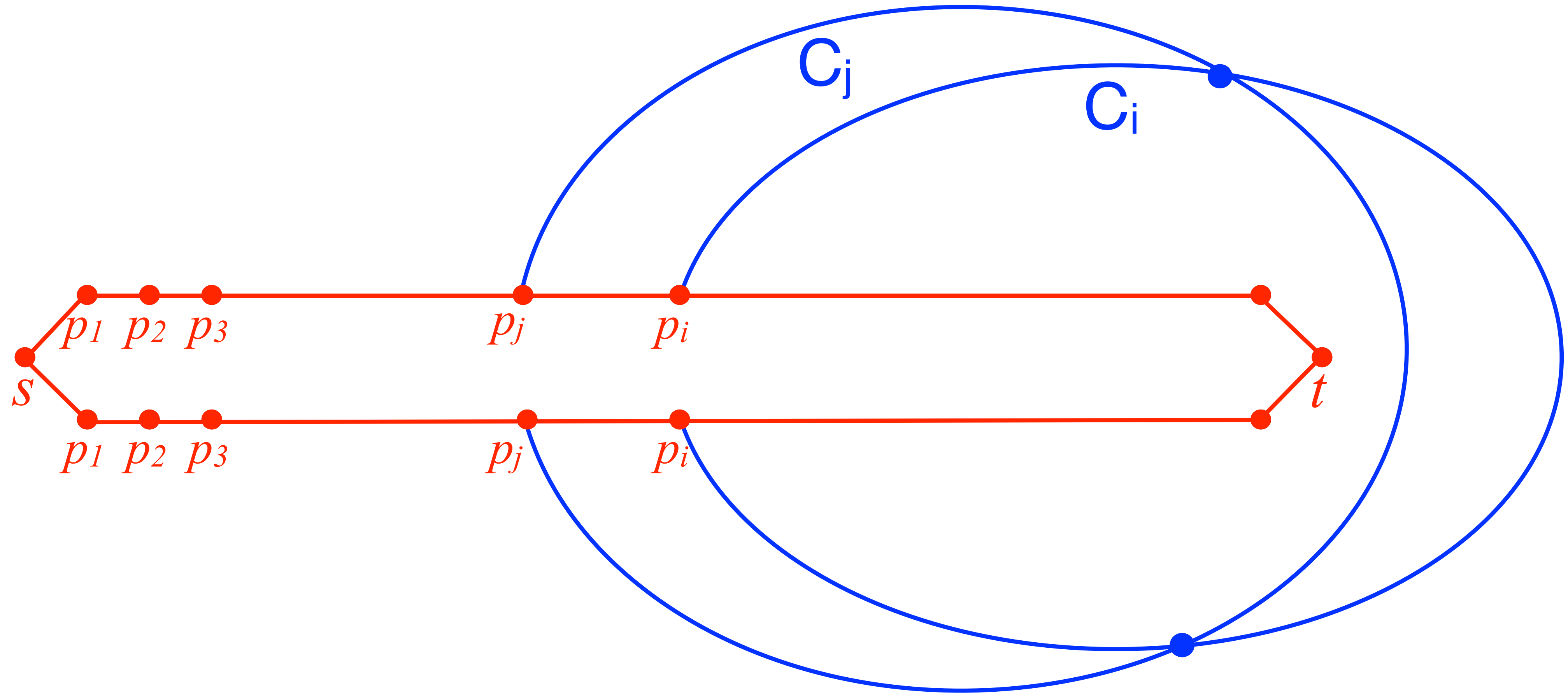
Undirected: Reif's Algorithm [Reif 1983]



SSSP computation in O(n) time [Henzinger et al 1997]

Correctness: cycles do not cross

Undirected: Reif's Algorithm [Reif 1983]

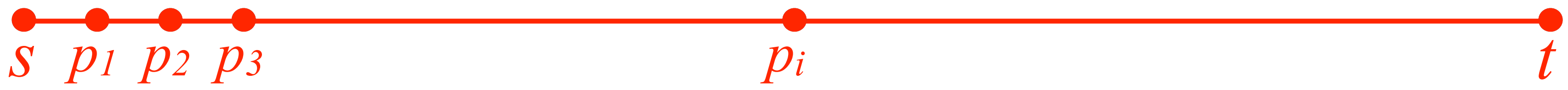


SSSP computation in O(n) time [Henzinger et al 1997]

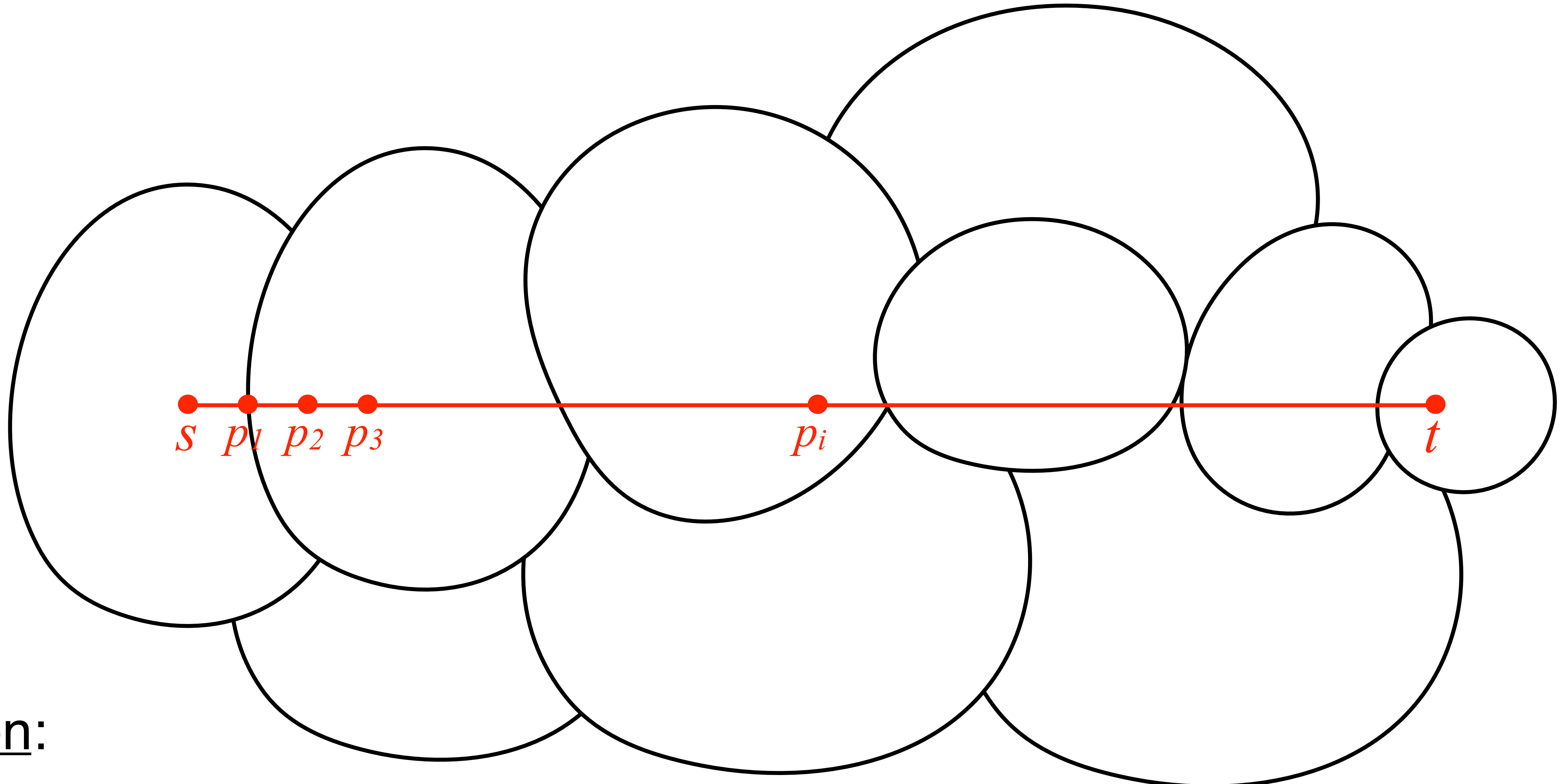
<u>Correctness</u>: cycles do not cross

<u>Time</u>: O(nlogn)

Coarse Reif [Italiano,Nussbaum,Sankowski,Wulff-Nilsen STOC 2011]



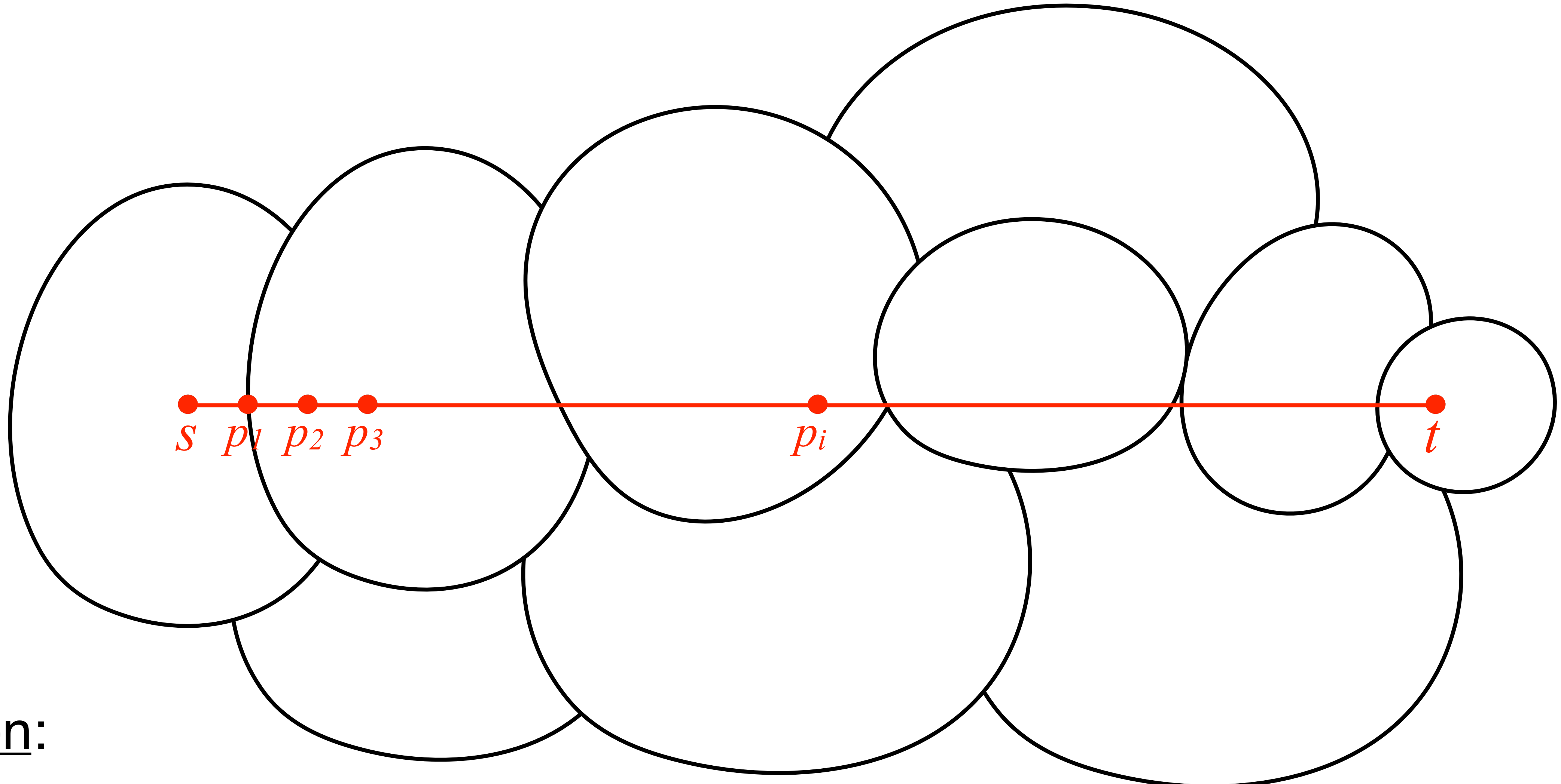
Coarse Reif [Italiano,Nussbaum,Sankowski,Wulff-Nilsen STOC 2011]



r-division:

O(n/r) pieces each with O(r) vertices and O(√r) boundary vertices

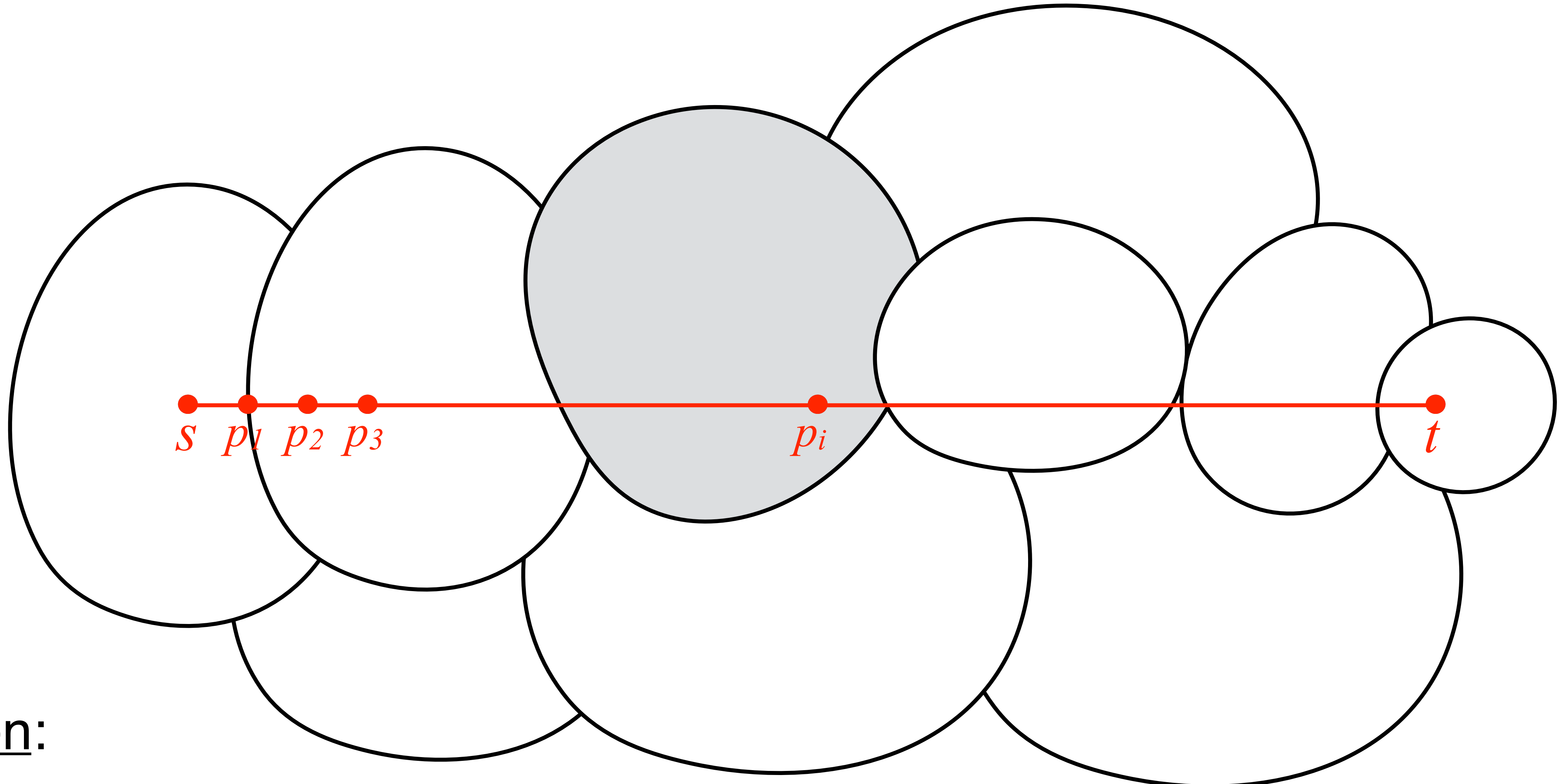
Coarse Reif [Italiano,Nussbaum,Sankowski,Wulff-Nilsen STOC 2011]



r-division:

O(n/r) pieces each with O(r) vertices and O(√r) boundary vertices

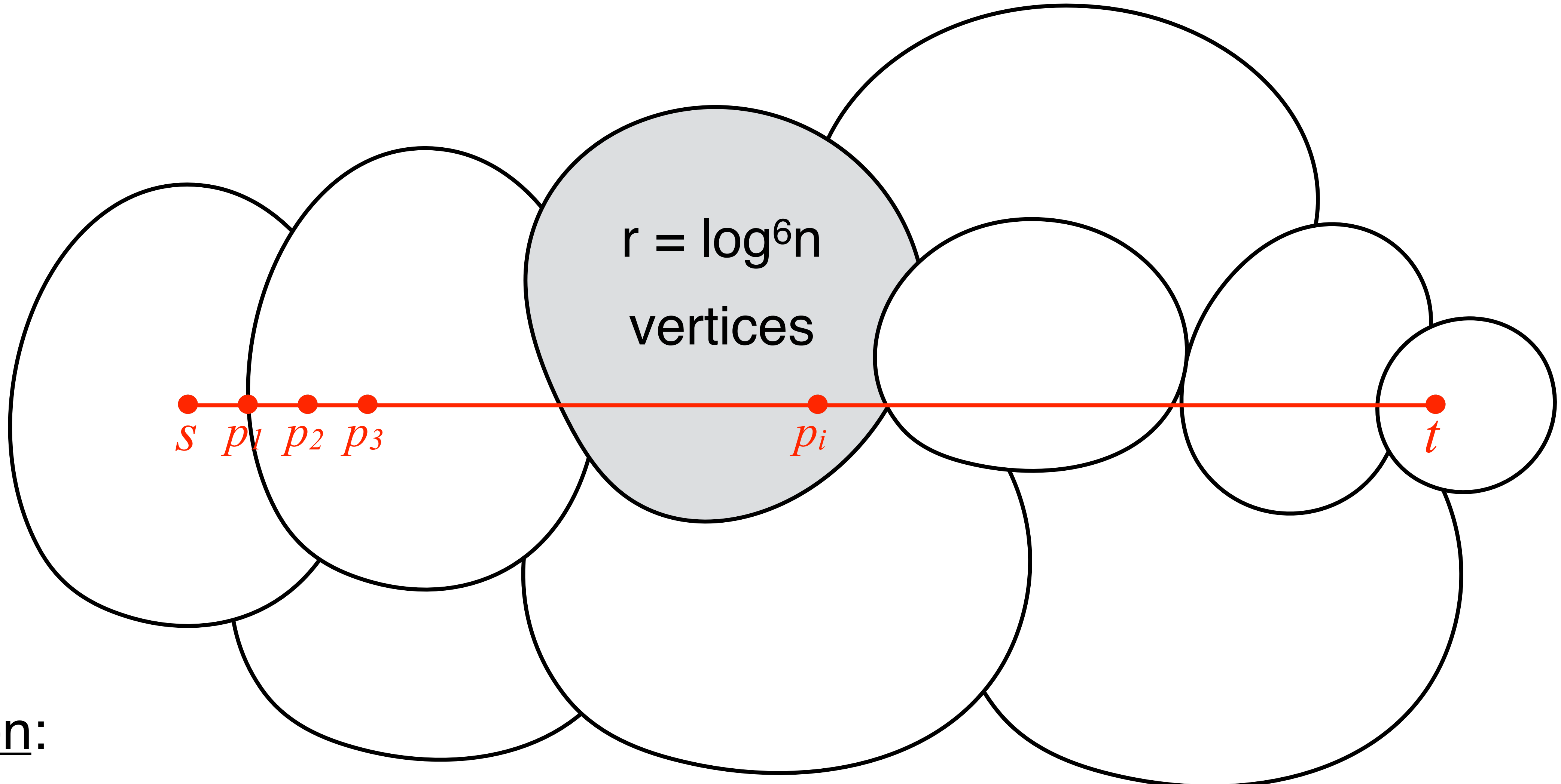
Coarse Reif [Italiano,Nussbaum,Sankowski,Wulff-Nilsen STOC 2011]



<u>r-division</u>:

O(n/r) pieces each with O(r) vertices and O(√r) boundary vertices

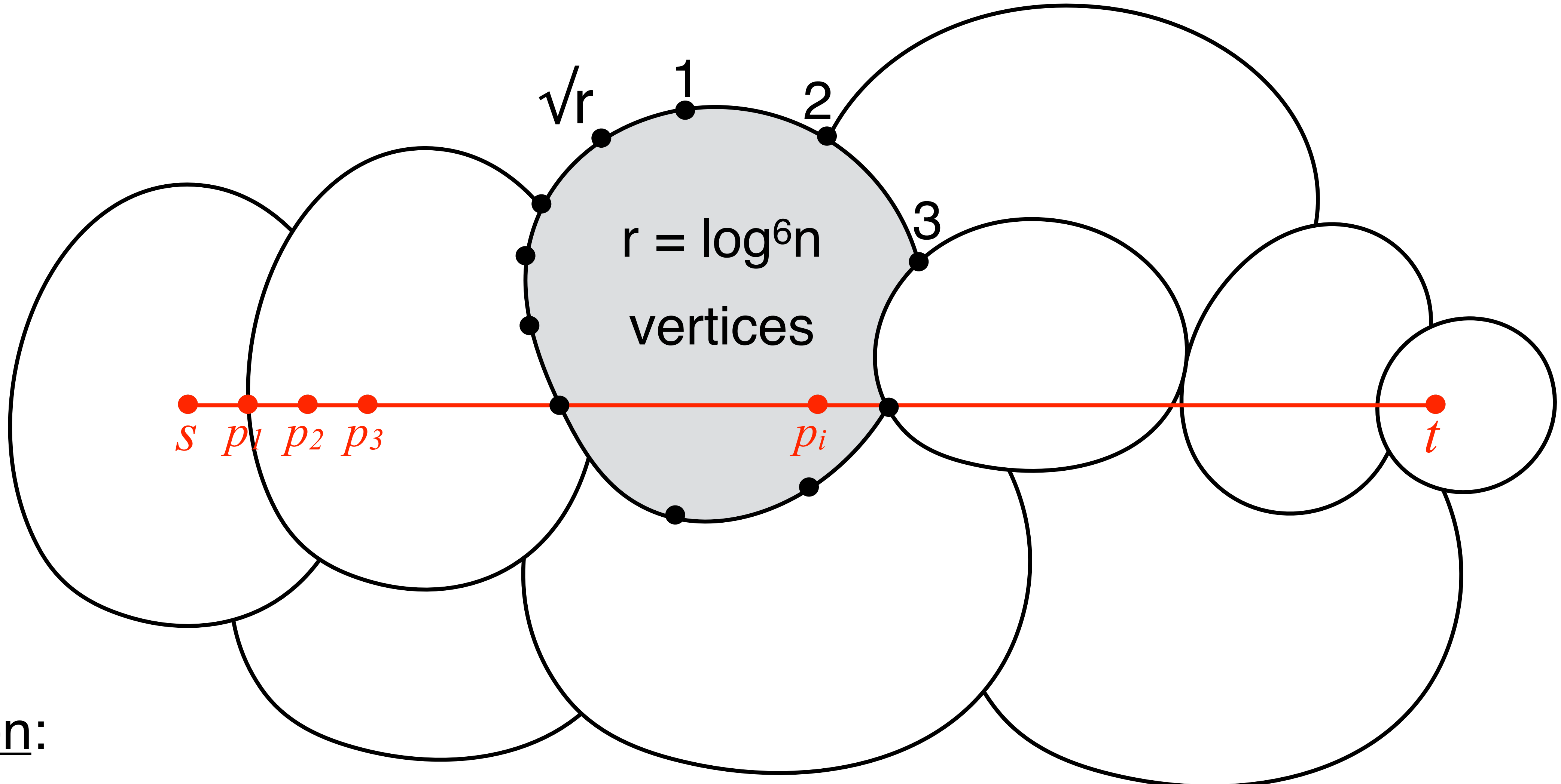
Coarse Reif [Italiano,Nussbaum,Sankowski,Wulff-Nilsen STOC 2011]



r-division:

O(n/r) pieces each with O(r) vertices and O(√r) boundary vertices

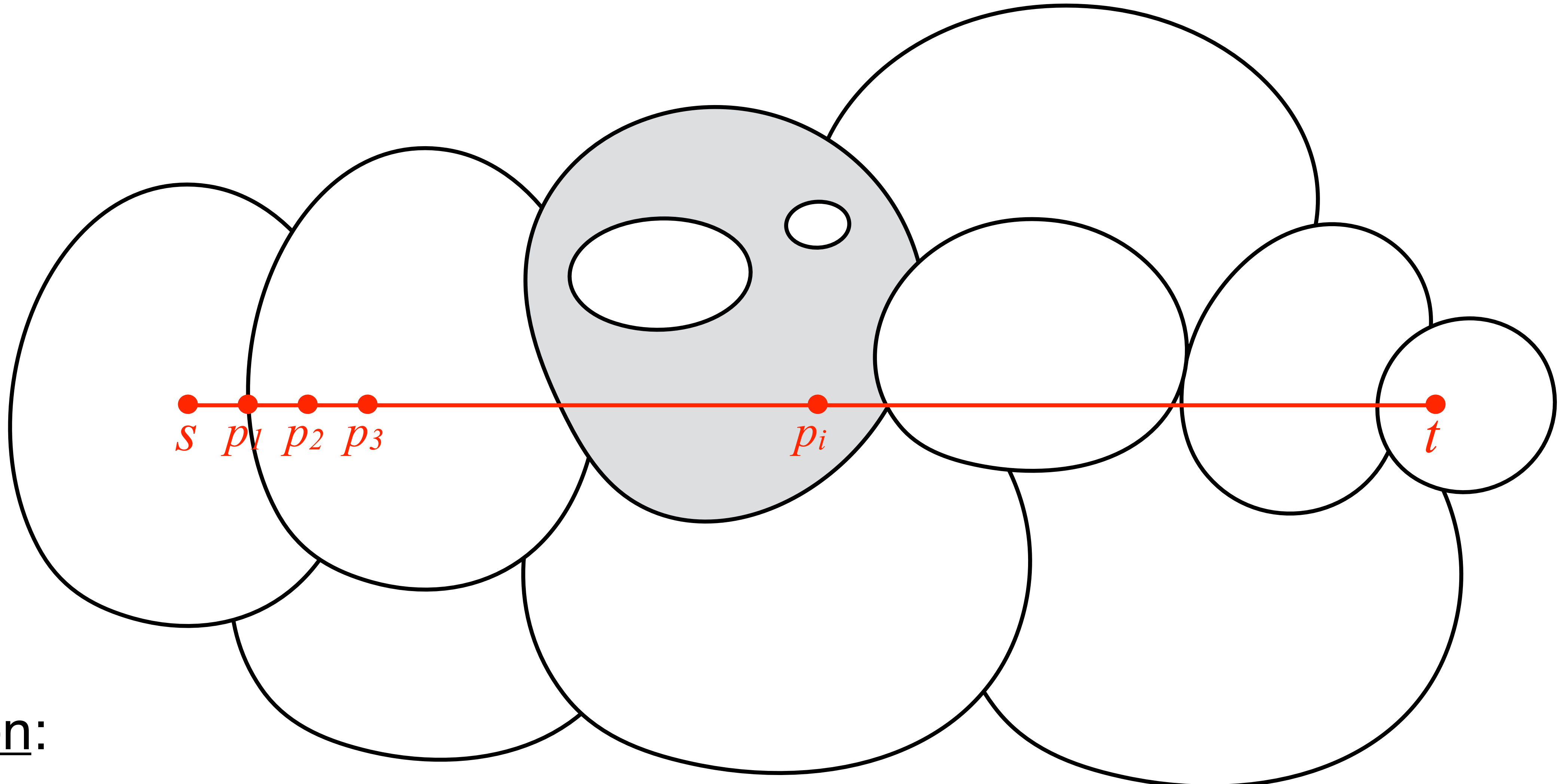
Coarse Reif [Italiano,Nussbaum,Sankowski,Wulff-Nilsen STOC 2011]



r-division:

O(n/r) pieces each with O(r) vertices and O(√r) boundary vertices

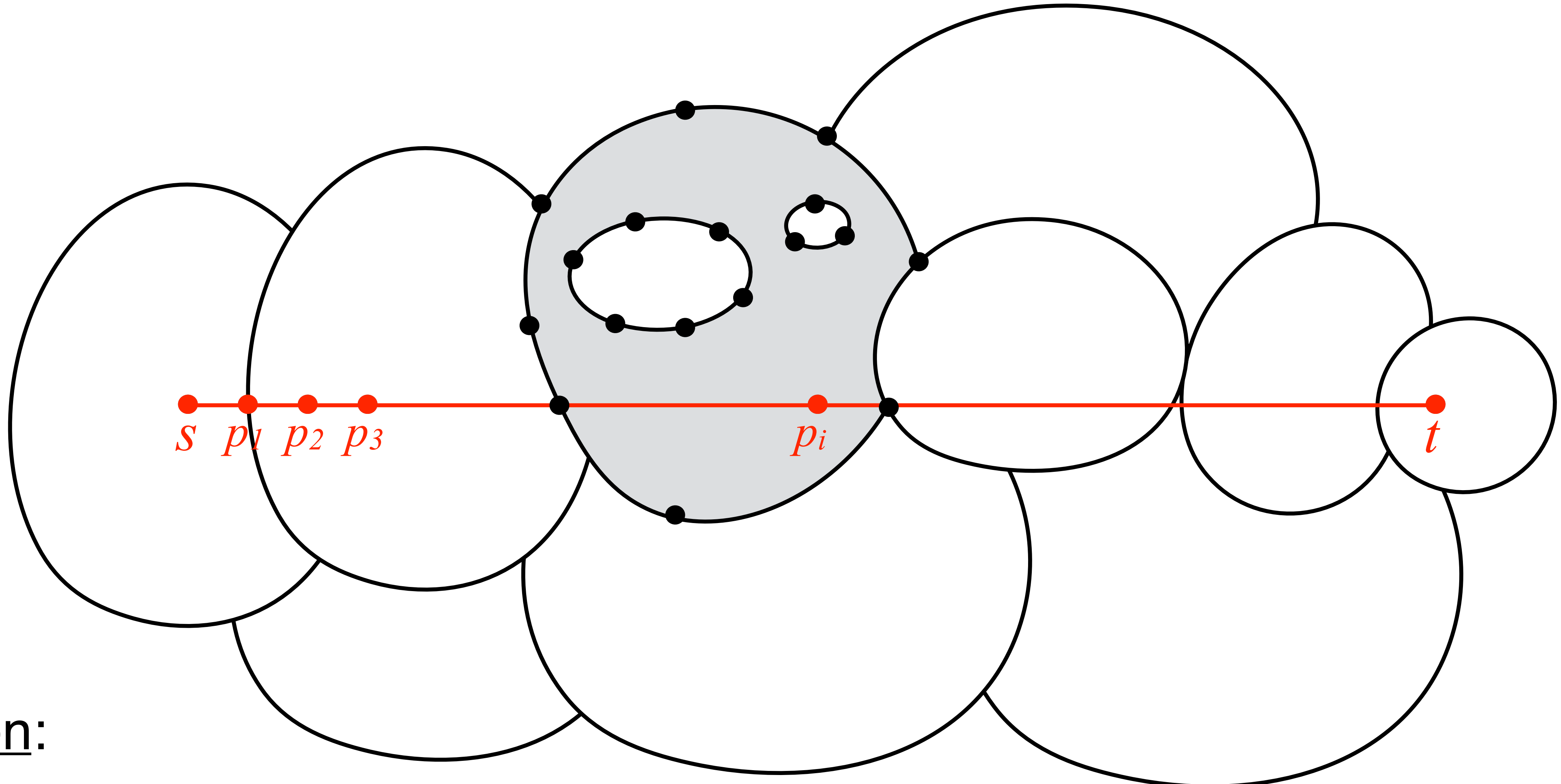
Coarse Reif [Italiano,Nussbaum,Sankowski,Wulff-Nilsen STOC 2011]



r-division:

O(n/r) pieces each with O(r) vertices and O(√r) boundary vertices

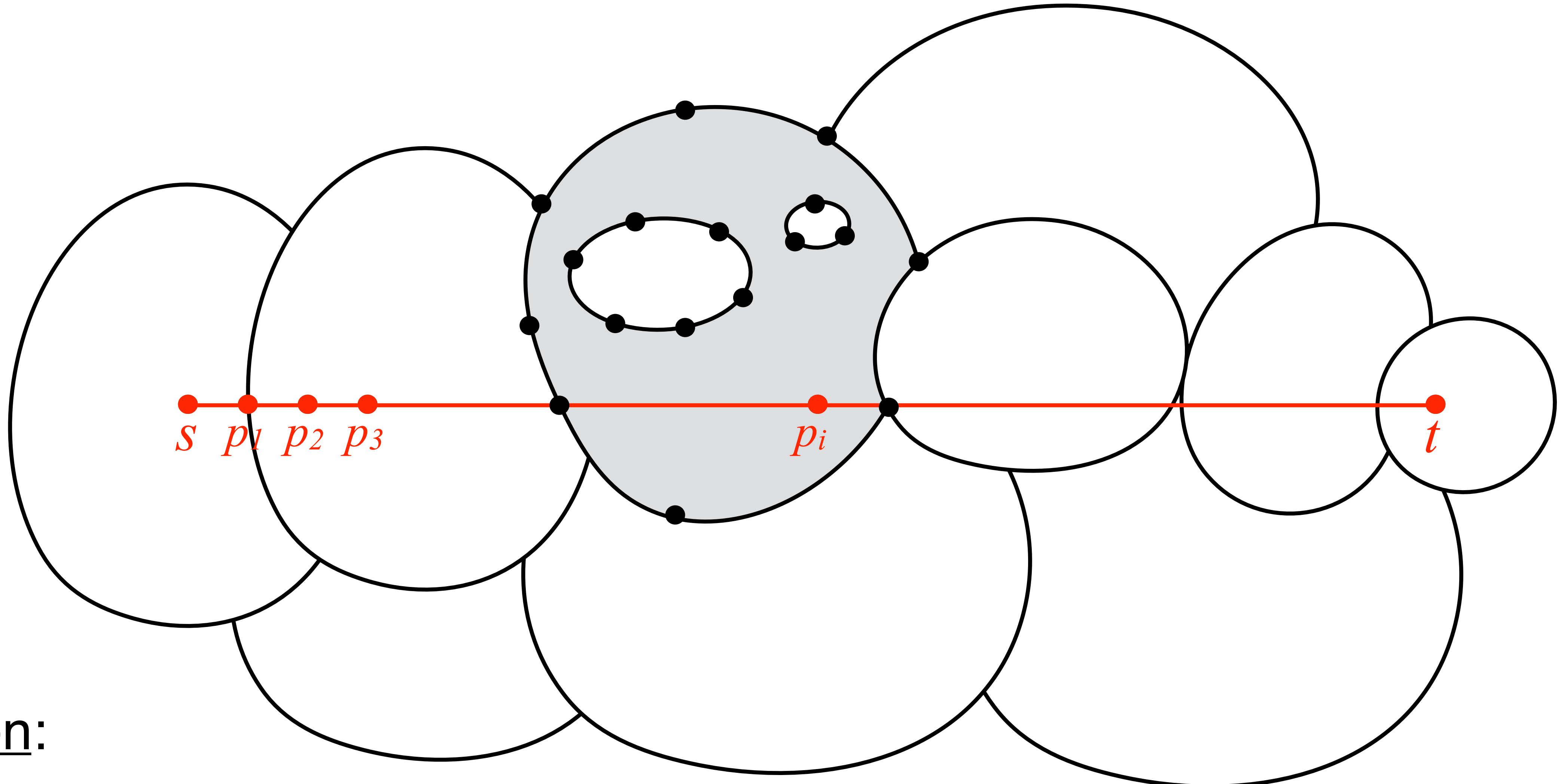
Coarse Reif [Italiano,Nussbaum,Sankowski,Wulff-Nilsen STOC 2011]



r-division:

O(n/r) pieces each with O(r) vertices and O(√r) boundary vertices

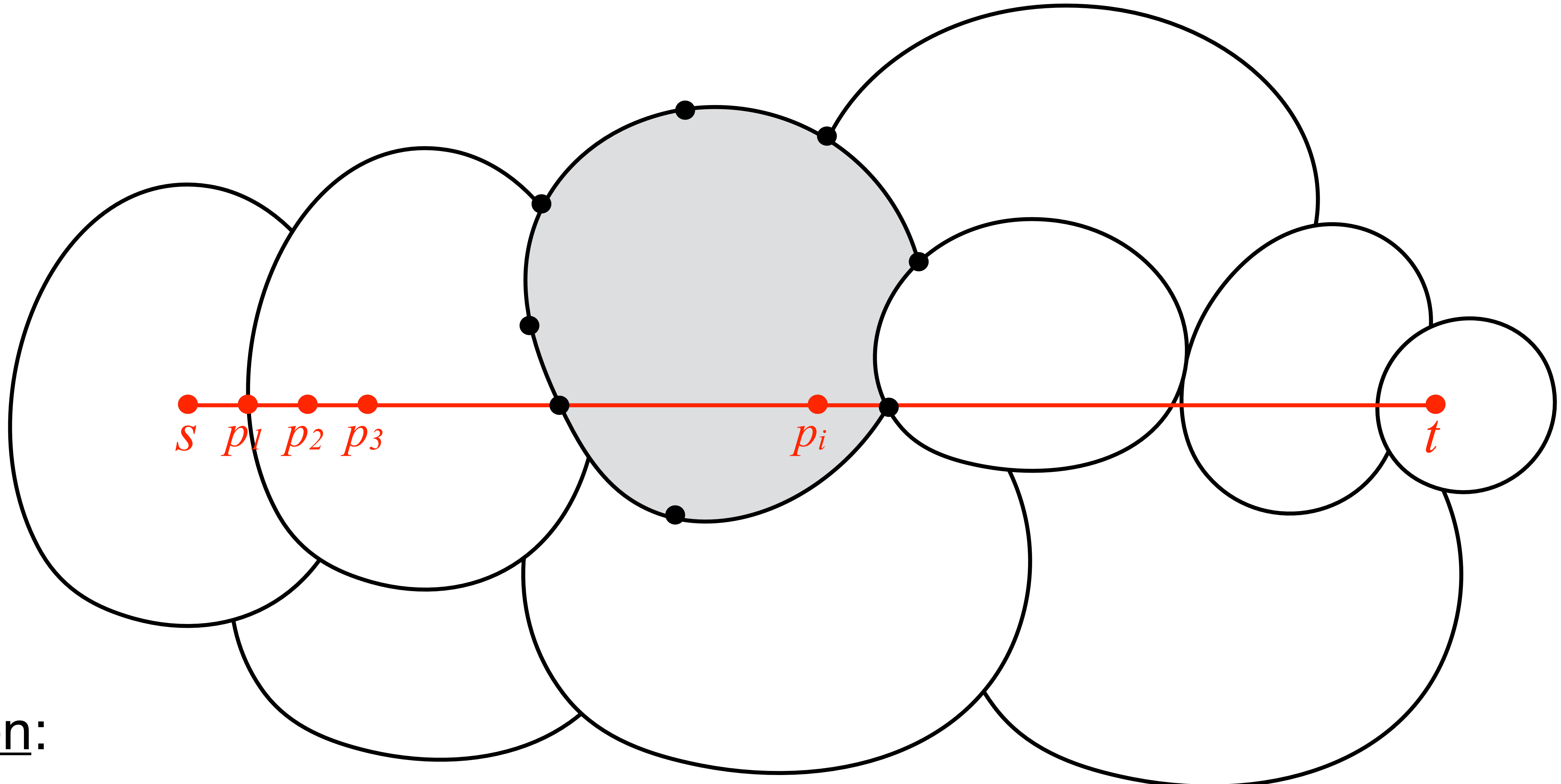
Coarse Reif [Italiano,Nussbaum,Sankowski,Wulff-Nilsen STOC 2011]



r-division:

O(n/r) pieces each with O(r) vertices and O(√r) boundary vertices with O(1) holes in O(n) time [Klein, Mozes, Sommer STOC 2013]

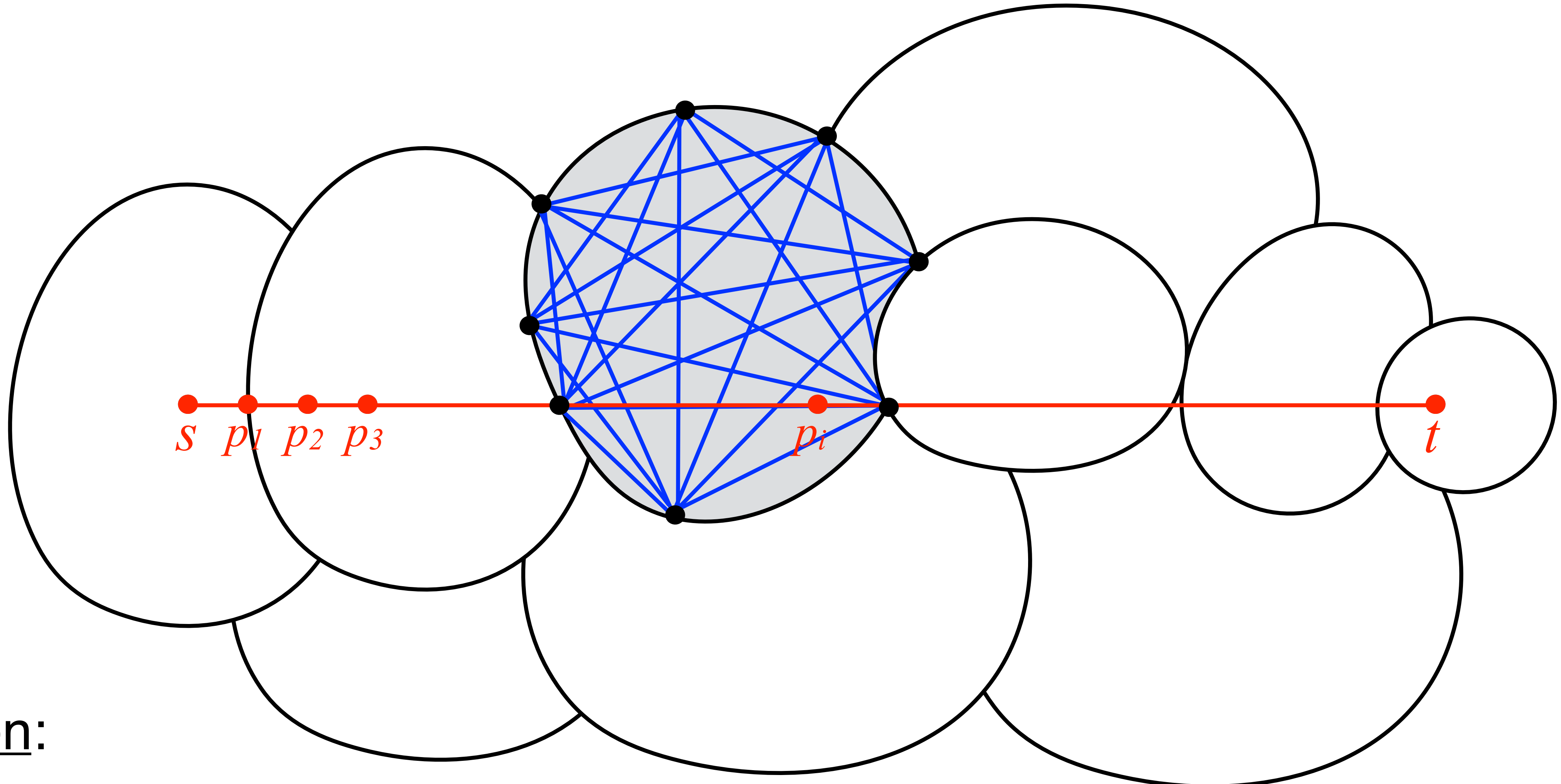
Coarse Reif [Italiano,Nussbaum,Sankowski,Wulff-Nilsen STOC 2011]



<u>r-division</u>:

O(n/r) pieces each with O(r) vertices and O(√r) boundary vertices with O(1) holes in O(n) time [Klein, Mozes, Sommer STOC 2013]

Coarse Reif [Italiano,Nussbaum,Sankowski,Wulff-Nilsen STOC 2011]



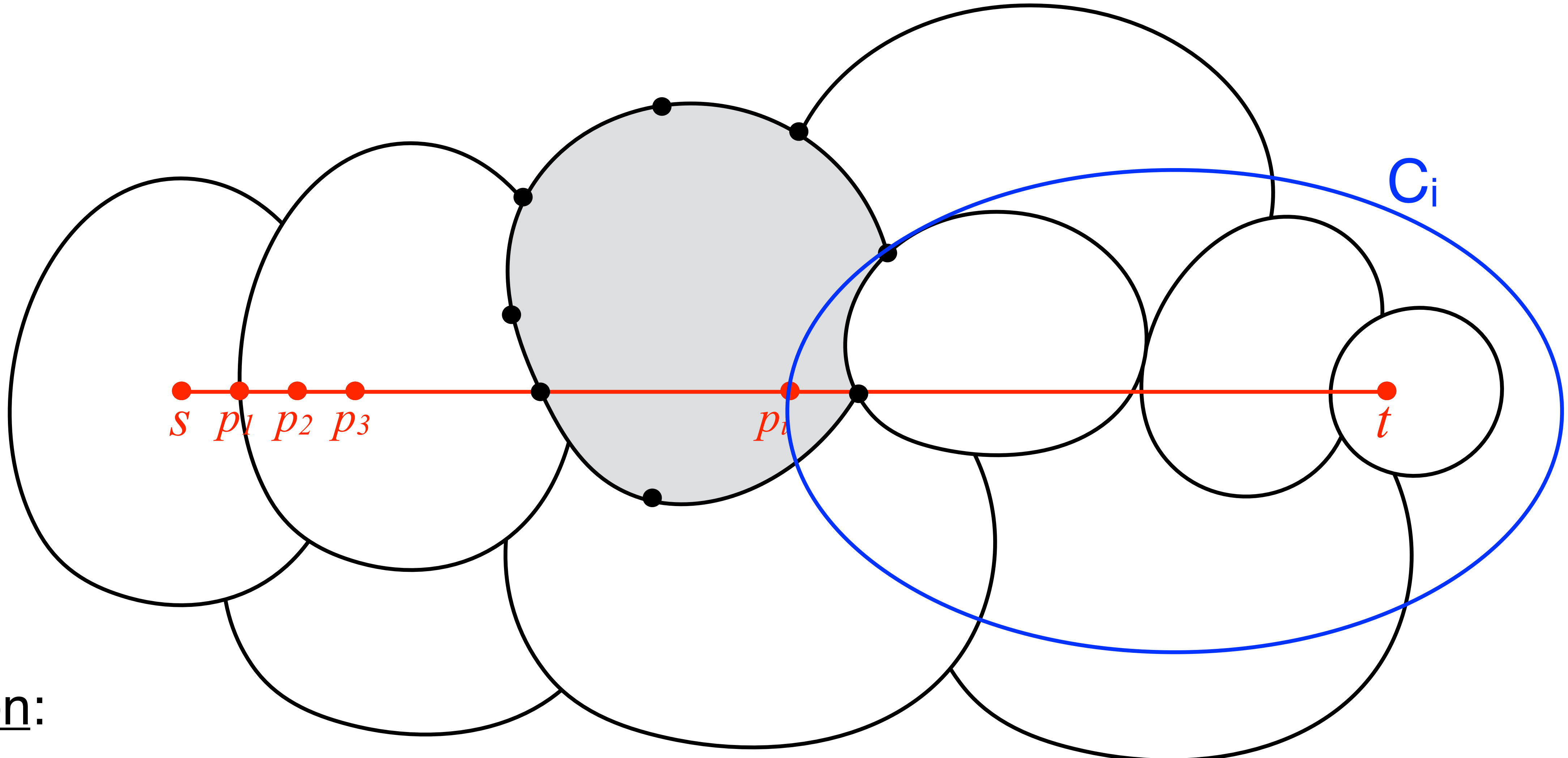
r-division:

O(n/r) pieces each with O(r) vertices and O(√r) boundary vertices
with O(1) holes in O(n) time [Klein, Mozes, Sommer STOC 2013]

Dense Distance Graph (DDG): all boundary-to-boundary distance matrices

O(nlogr) = O(nloglogn) time [Klein SODA 2005]

Coarse Reif [Italiano,Nussbaum,Sankowski,Wulff-Nilsen STOC 2011]



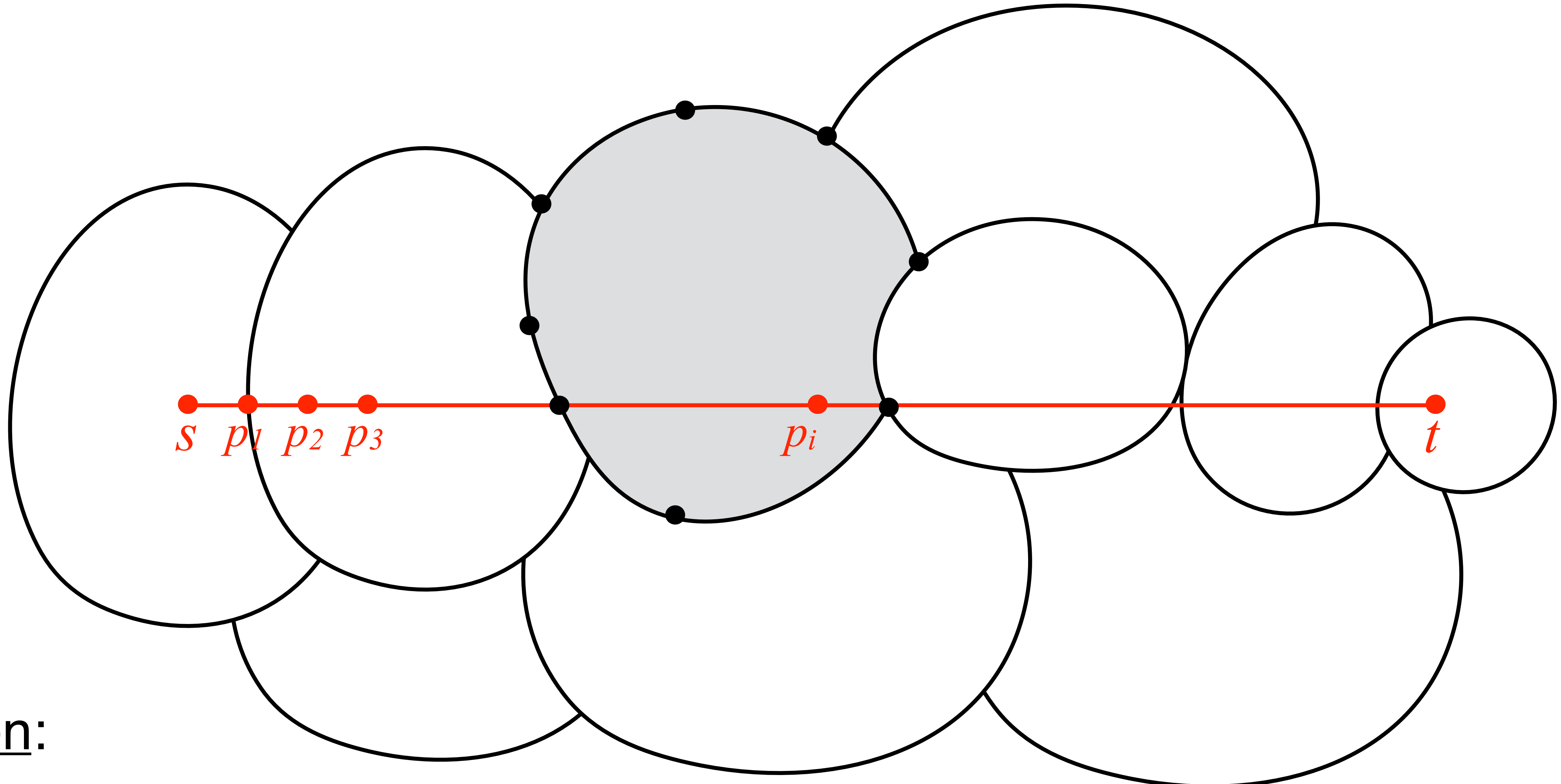
r-division:

O(n/r) pieces each with O(r) vertices and O(√r) boundary vertices
with O(1) holes in O(n) time [Klein, Mozes, Sommer STOC 2013]

Dense Distance Graph (DDG): all boundary-to-boundary distance matrices

O(nlogr) = O(nloglogn) time [Klein SODA 2005]

Coarse Reif [Italiano,Nussbaum,Sankowski,Wulff-Nilsen STOC 2011]



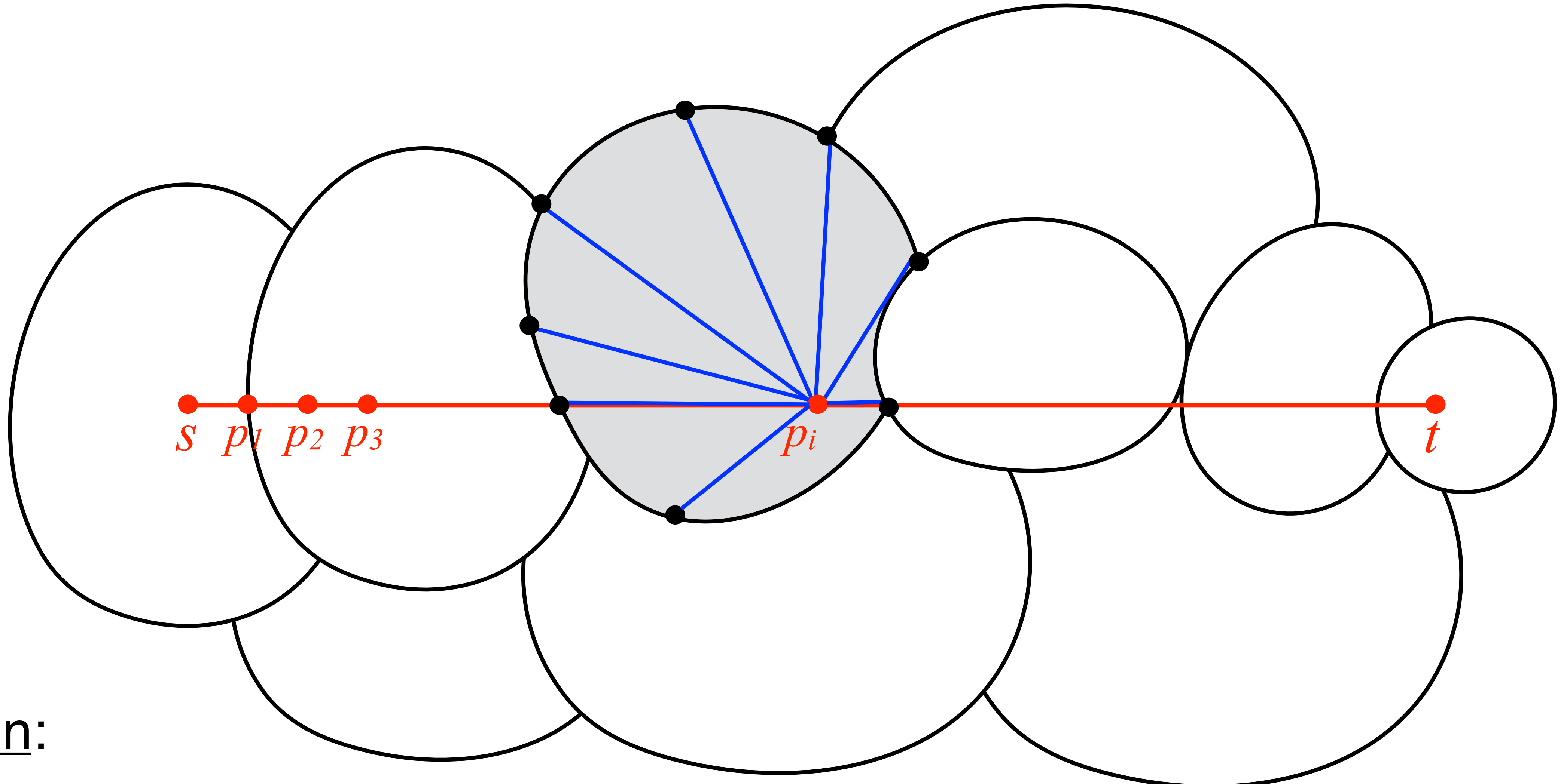
r-division:

O(n/r) pieces each with O(r) vertices and O(√r) boundary vertices
with O(1) holes in O(n) time [Klein, Mozes, Sommer STOC 2013]

Dense Distance Graph (DDG): all boundary-to-boundary distance matrices

O(nlogr) = O(nloglogn) time [Klein SODA 2005]

Coarse Reif [Italiano,Nussbaum,Sankowski,Wulff-Nilsen STOC 2011]



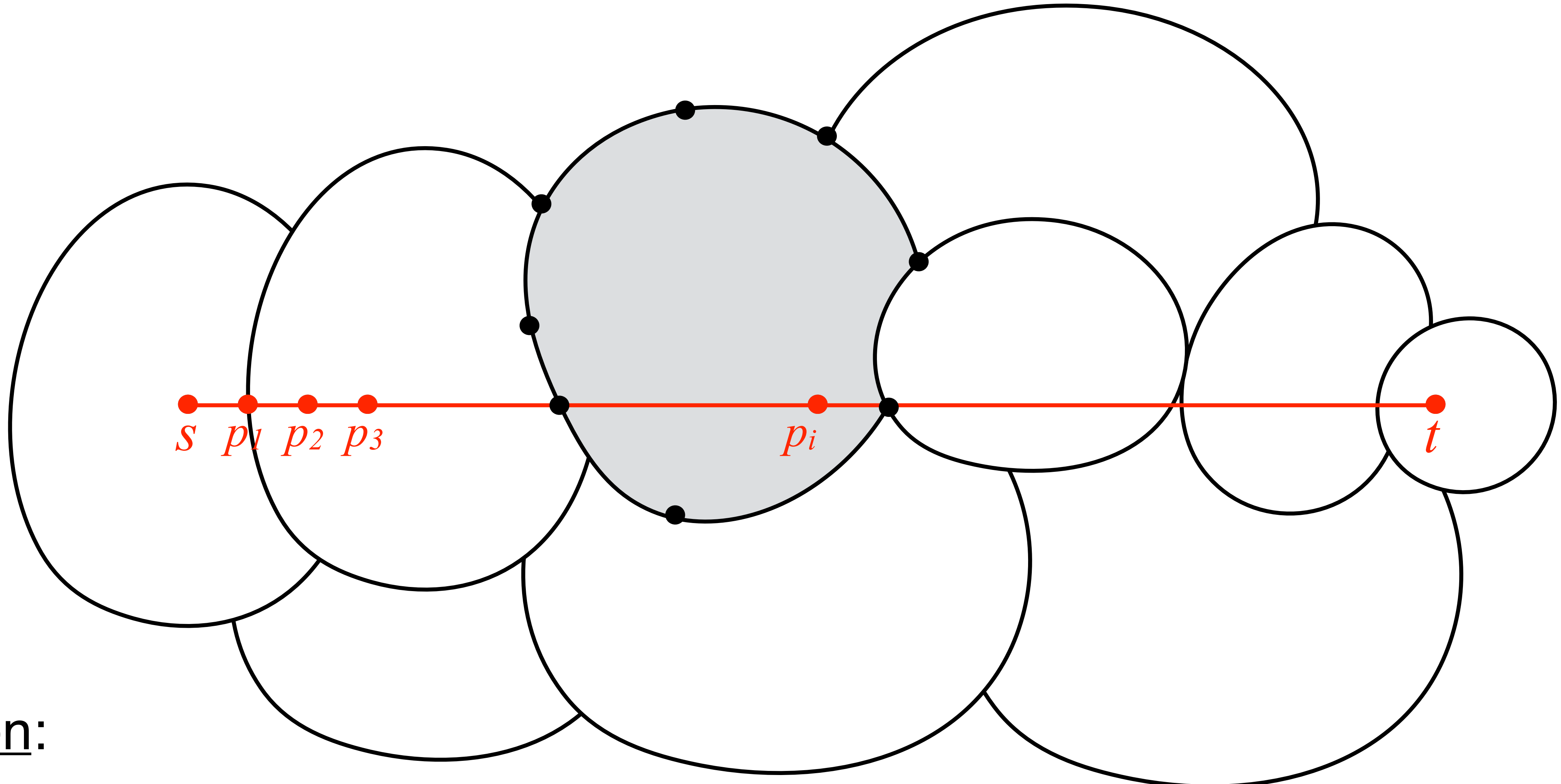
r-division:

O(n/r) pieces each with O(r) vertices and O(√r) boundary vertices with O(1) holes in O(n) time [Klein, Mozes, Sommer STOC 2013]

Dense Distance Graph (DDG): all boundary-to-boundary distance matrices

O(nlogr) = O(nloglogn) time [Klein SODA 2005]

Coarse Reif [Italiano,Nussbaum,Sankowski,Wulff-Nilsen STOC 2011]



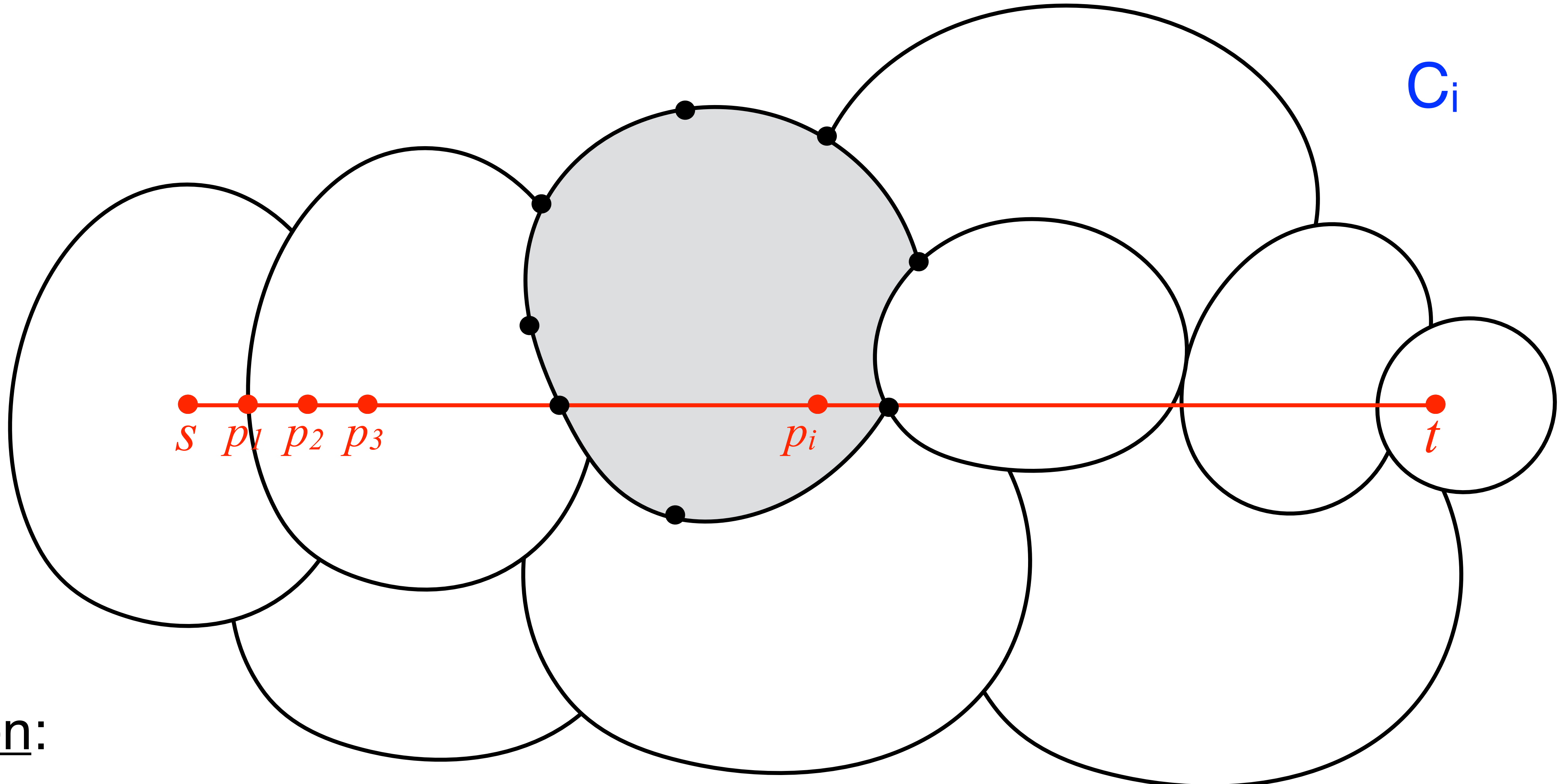
r-division:

O(n/r) pieces each with O(r) vertices and O(√r) boundary vertices with O(1) holes in O(n) time [Klein, Mozes, Sommer STOC 2013]

Dense Distance Graph (DDG): all boundary-to-boundary distance matrices

O(nlogr) = O(nloglogn) time [Klein SODA 2005]

Coarse Reif [Italiano,Nussbaum,Sankowski,Wulff-Nilsen STOC 2011]



r-division:

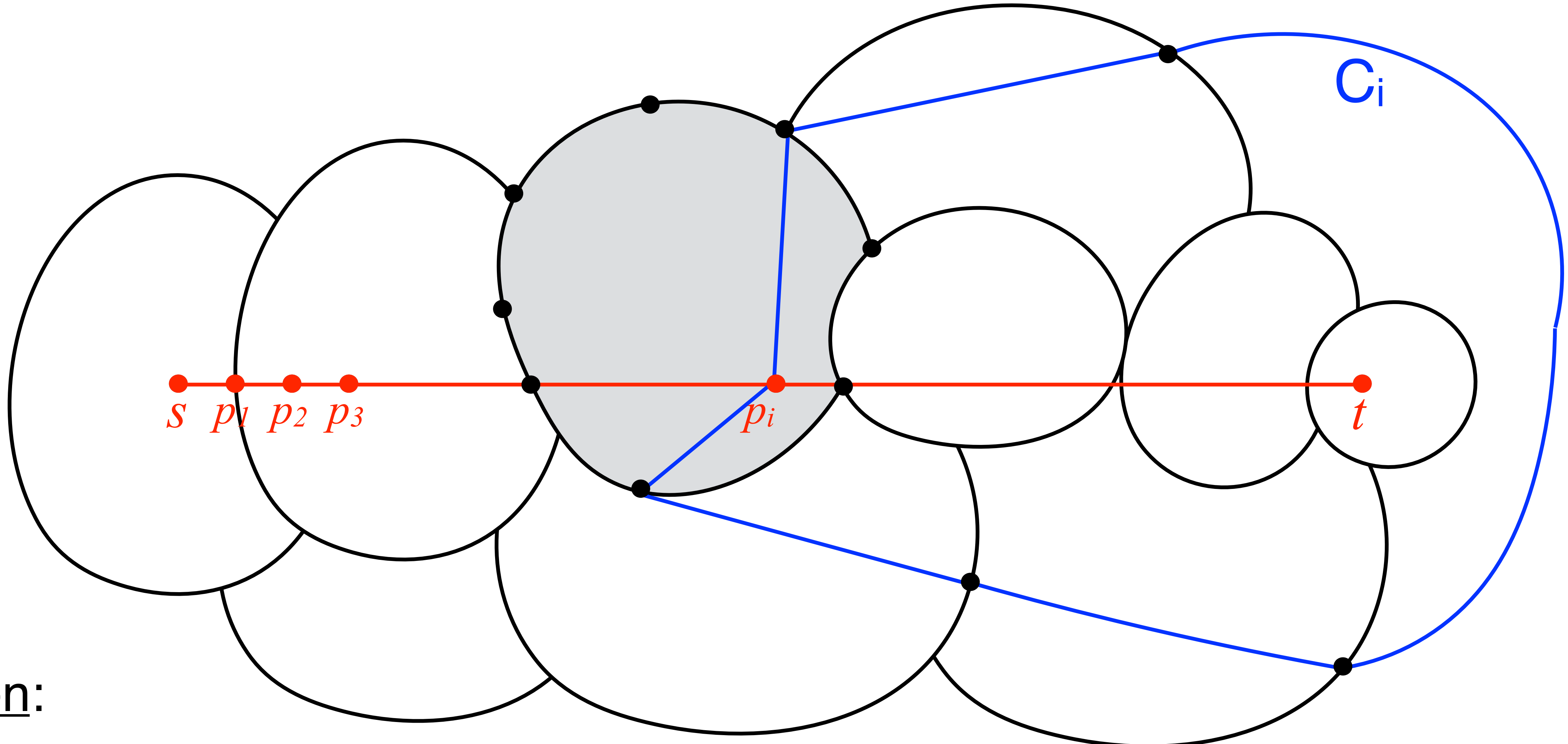
O(n/r) pieces each with O(r) vertices and O(√r) boundary vertices with O(1) holes in O(n) time [Klein, Mozes, Sommer STOC 2013]

Dense Distance Graph (DDG): all boundary-to-boundary distance matrices

O(nlogr) = O(nloglogn) time [Klein SODA 2005]

Find C_i with SSSP on DDG:

Coarse Reif [Italiano,Nussbaum,Sankowski,Wulff-Nilsen STOC 2011]



<u>r-division</u>:

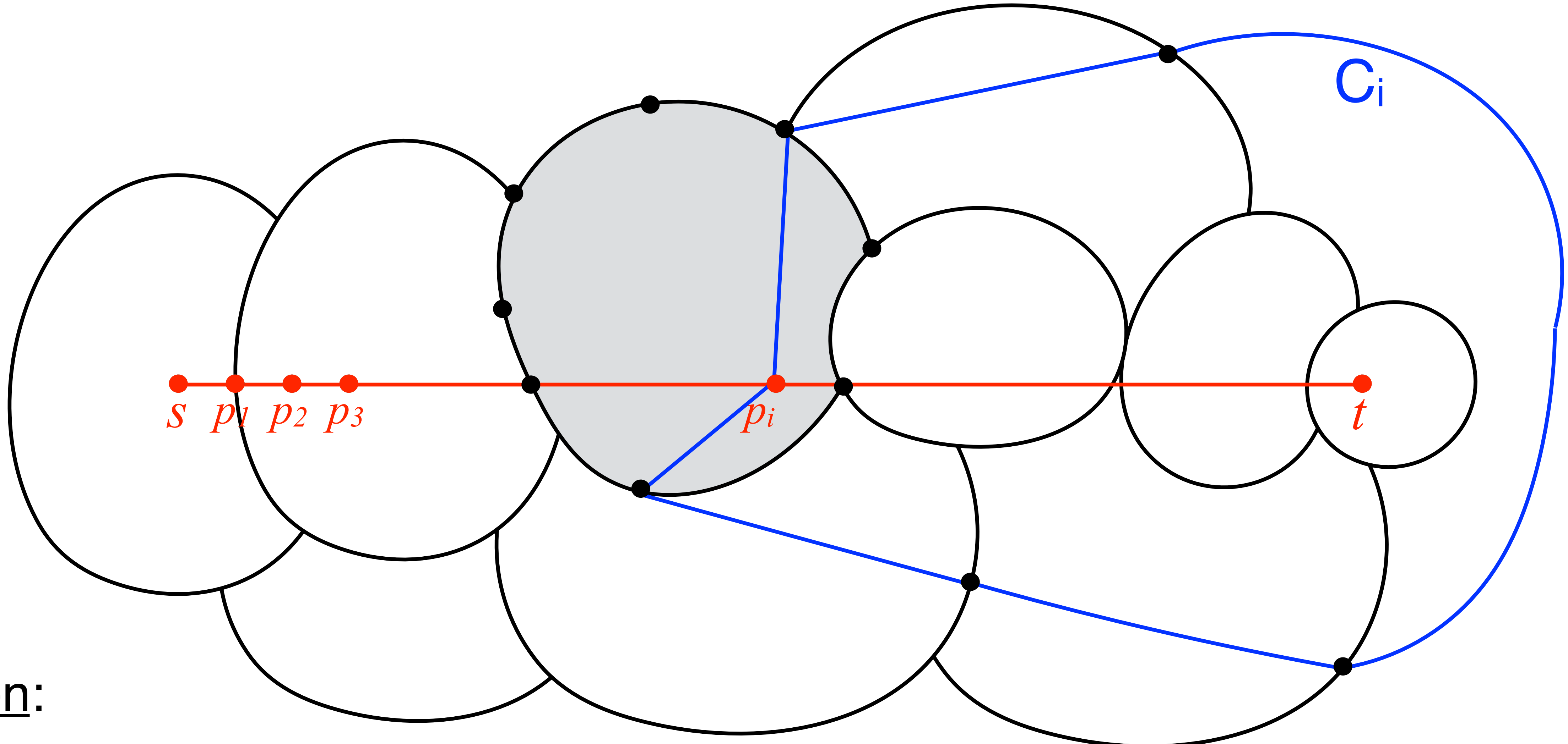
O(n/r) pieces each with O(r) vertices and O(√r) boundary vertices
with O(1) holes in O(n) time [Klein, Mozes, Sommer STOC 2013]

<u>Dense Distance Graph (DDG)</u>: all boundary-to-boundary distance matrices

O(nlogr) = O(nloglogn) time [Klein SODA 2005]

<u>Find C_i with SSSP on DDG</u>:

Coarse Reif [Italiano,Nussbaum,Sankowski,Wulff-Nilsen STOC 2011]



r-division:

O(n/r) pieces each with O(r) vertices and O(√r) boundary vertices with O(1) holes in O(n) time [Klein, Mozes, Sommer STOC 2013]

Dense Distance Graph (DDG): all boundary-to-boundary distance matrices

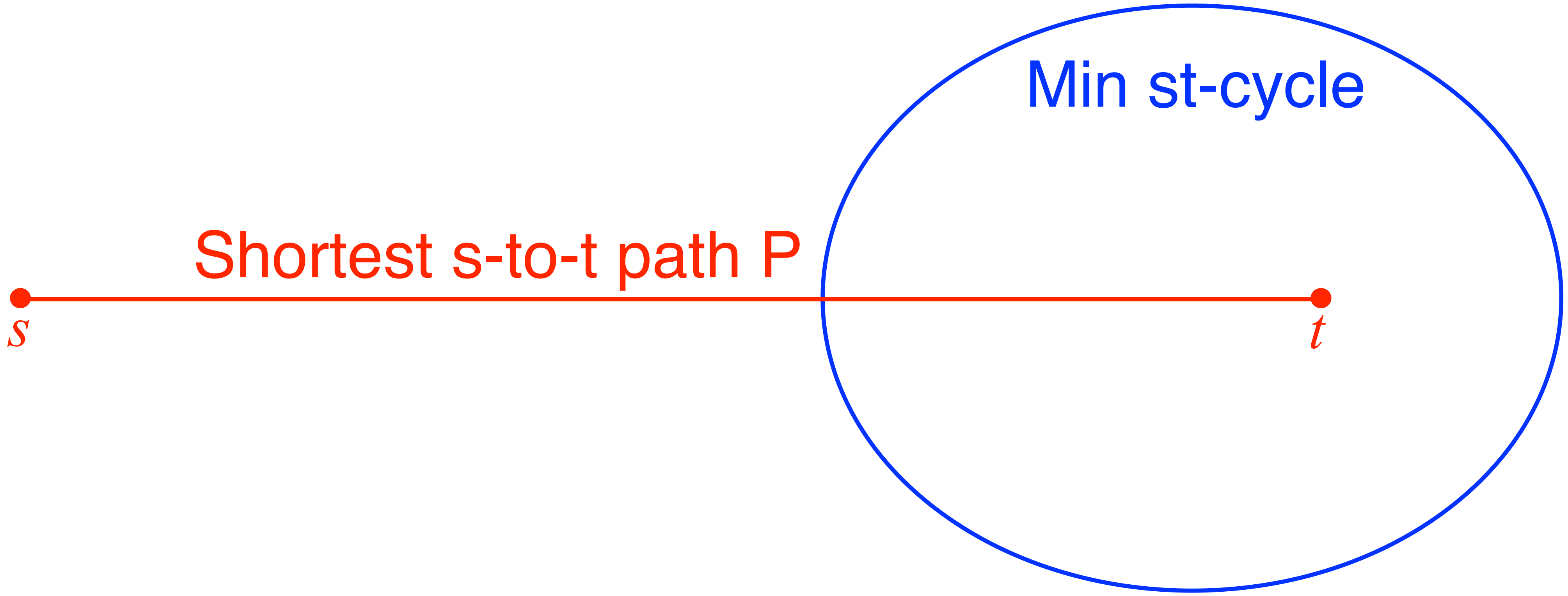
O(nlogr) = O(nloglogn) time [Klein SODA 2005]

Find C_i with SSSP on DDG:

$O(n\log^2 n/\sqrt{r}) = O(n/\log n)$ sublinear time [Fakcharoenphol, Rao FOCS 2001]

Undirected Min st-cycle in O(nloglogn) time

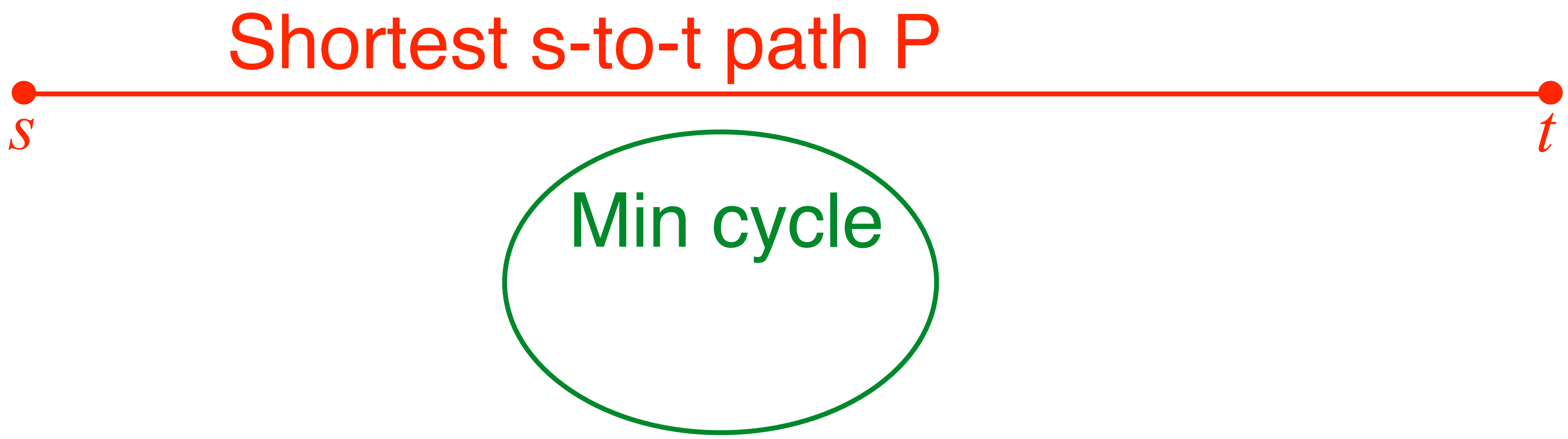
[Italiano,Nussbaum,Sankowski,Wulff-Nilsen STOC 2011]



Bottleneck: computing DDG (O(nloglogn) time)

Undirected Min cycle in O(nloglogn) time

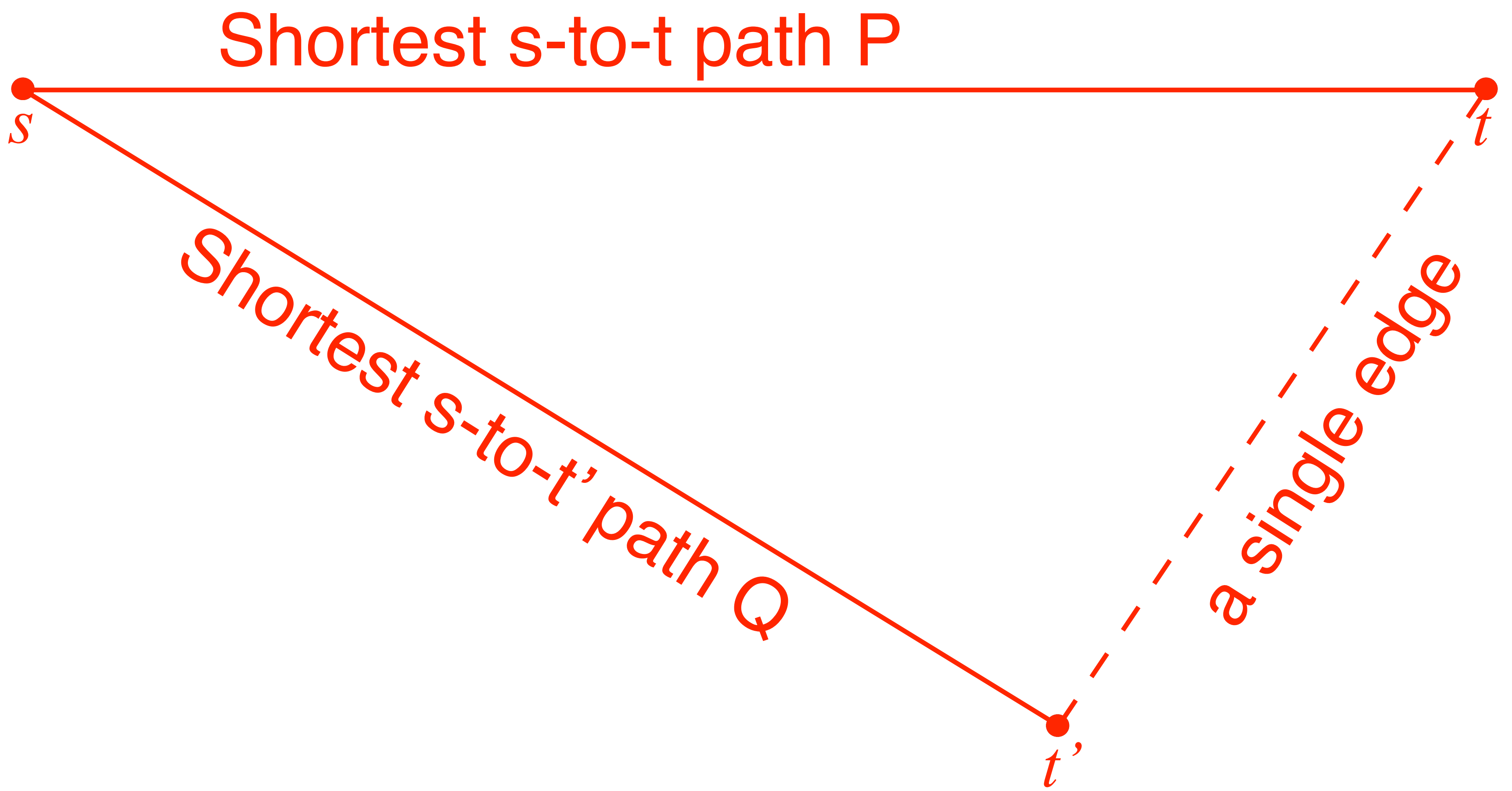
[Łącki, Sankowski ESA 2011]



Undirected Min cycle in O(nloglogn) time

[Łącki, Sankowski ESA 2011]

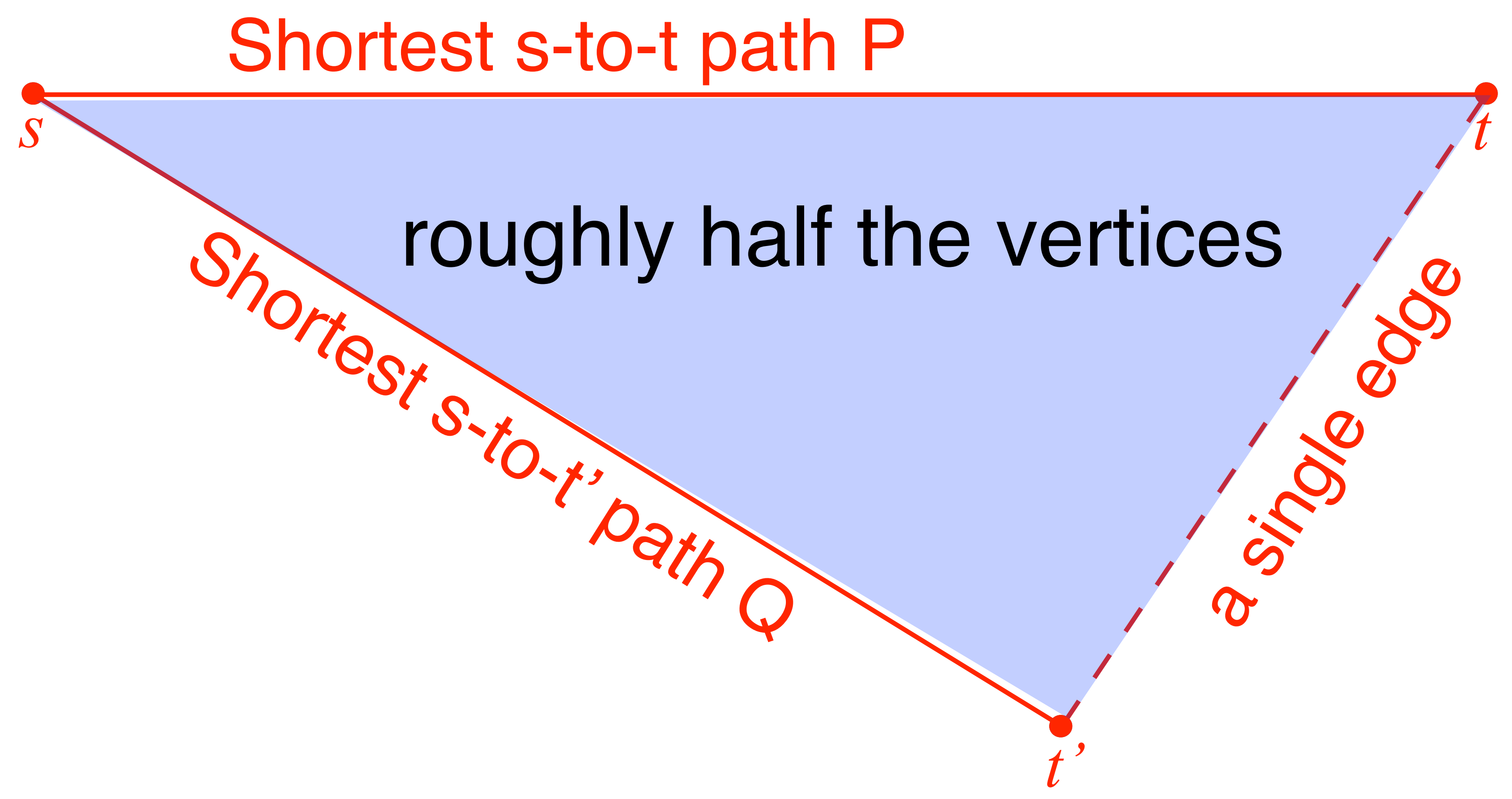
Shortest Path Separator



Undirected Min cycle in O(nloglogn) time

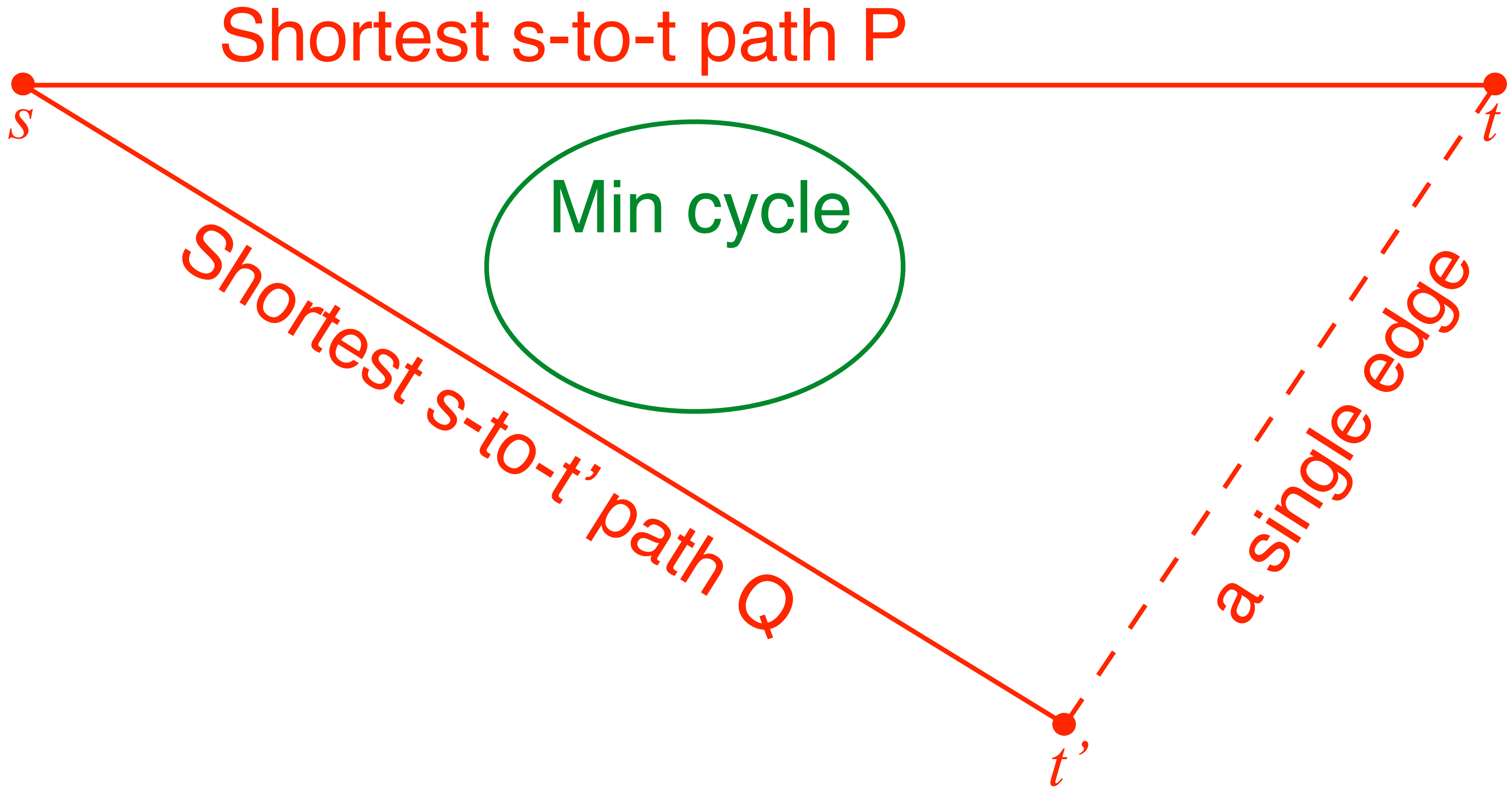
[Łącki, Sankowski ESA 2011]

Shortest Path Separator



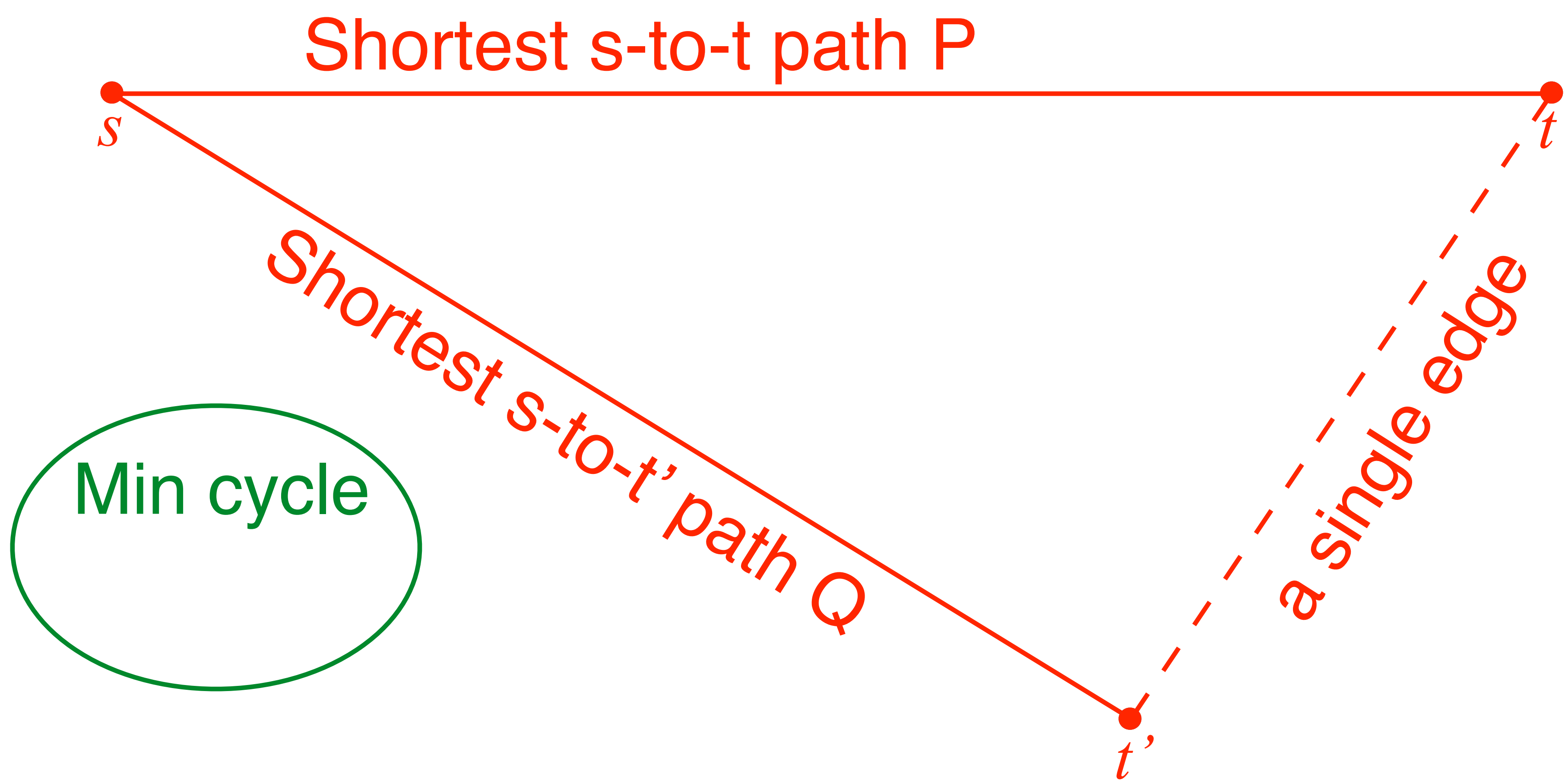
Undirected Min cycle in O(nloglogn) time

[Łącki, Sankowski ESA 2011]



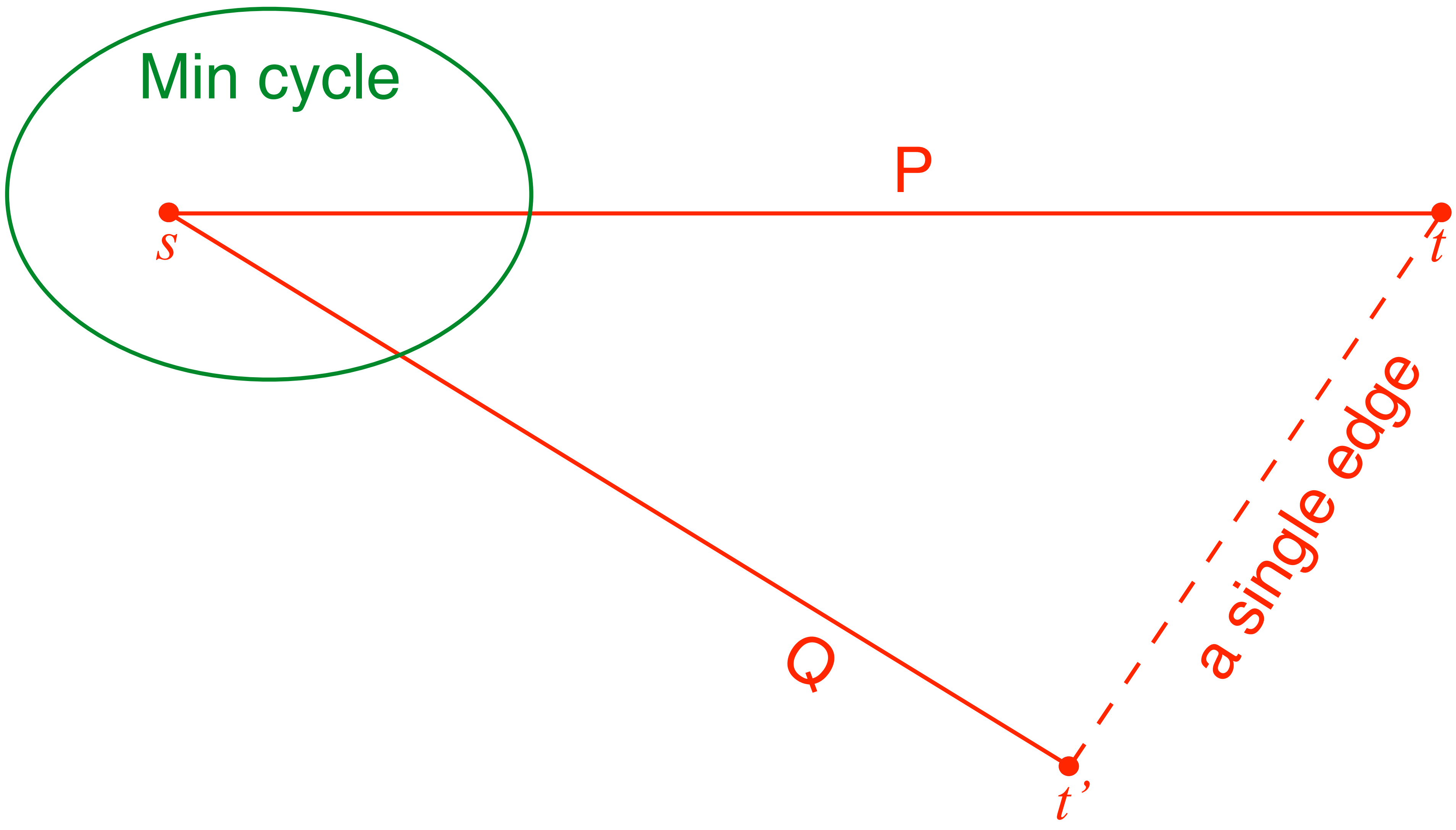
Undirected Min cycle in O(nloglogn) time

[Łącki, Sankowski ESA 2011]



Undirected Min cycle in O(nloglogn) time

[Łącki, Sankowski ESA 2011]



Undirected Min cycle in O(nloglogn) time

[Łącki, Sankowski ESA 2011]

Undirected Min cycle in O(nloglogn) time

[Łącki, Sankowski ESA 2011]

Find shortest cycle that crosses P once

Undirected Min cycle in O(nloglogn) time

[Łącki, Sankowski ESA 2011]

Find shortest cycle that crosses P once

We've just seen this takes O(nloglogn) time so O(n logn loglogn) overall

[Chalermsook, Fakcharoenphol, Nanongkai SODA 2004] +
[Italiano,Nussbaum,Sankowski,Wulff-Nilsen STOC 2011]

Undirected Min cycle in O(nloglogn) time

[Łącki, Sankowski ESA 2011]

Find shortest cycle that crosses P once

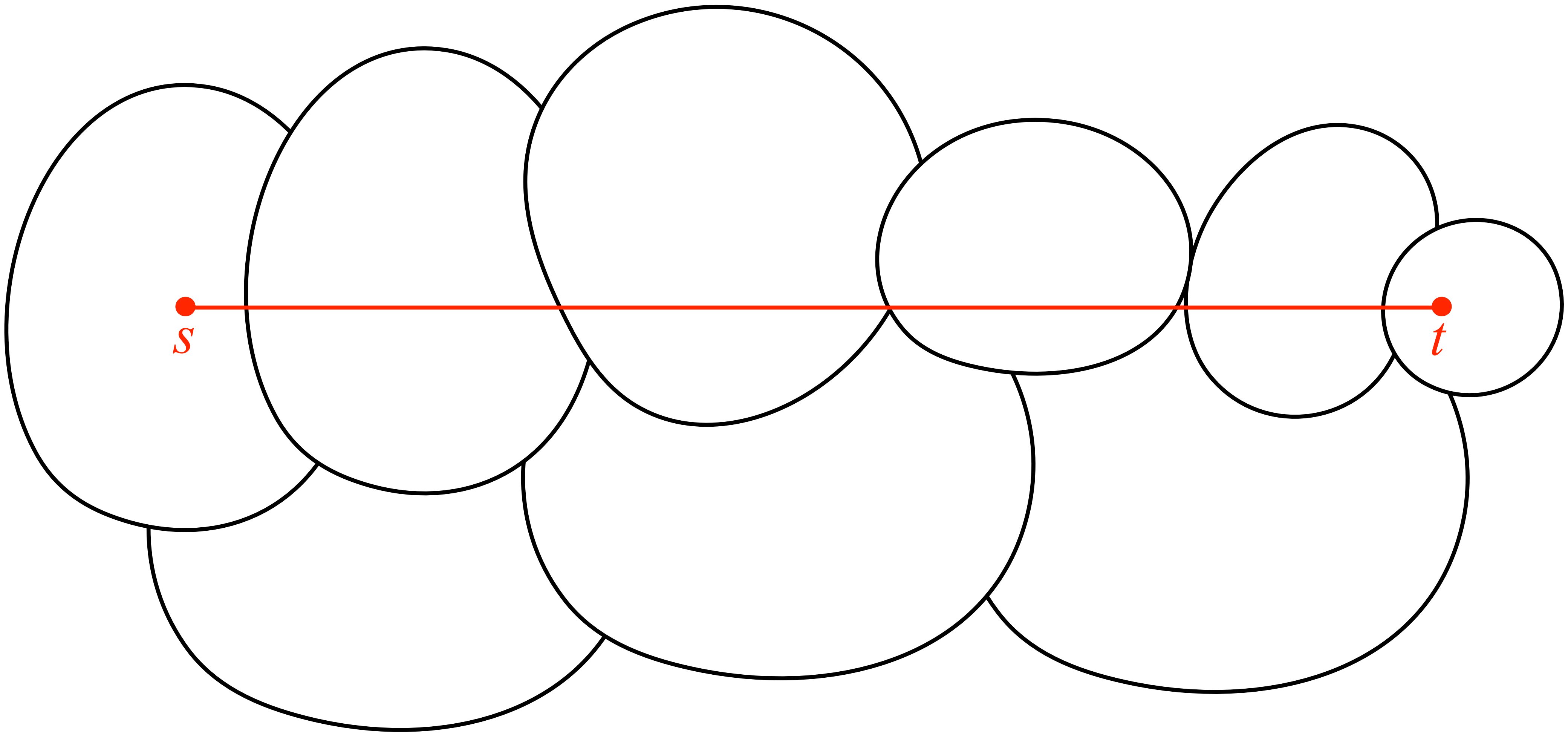
We've just seen this takes O(nloglogn) time so O(n logn loglogn) overall

[Chalermsook, Fakcharoenphol, Nanongkai SODA 2004] +
[Italiano,Nussbaum,Sankowski,Wulff-Nilsen STOC 2011]

Bottleneck: recomputing DDG in each recursive level (O(nloglogn) time)

Undirected Min cycle in O(nloglogn) time

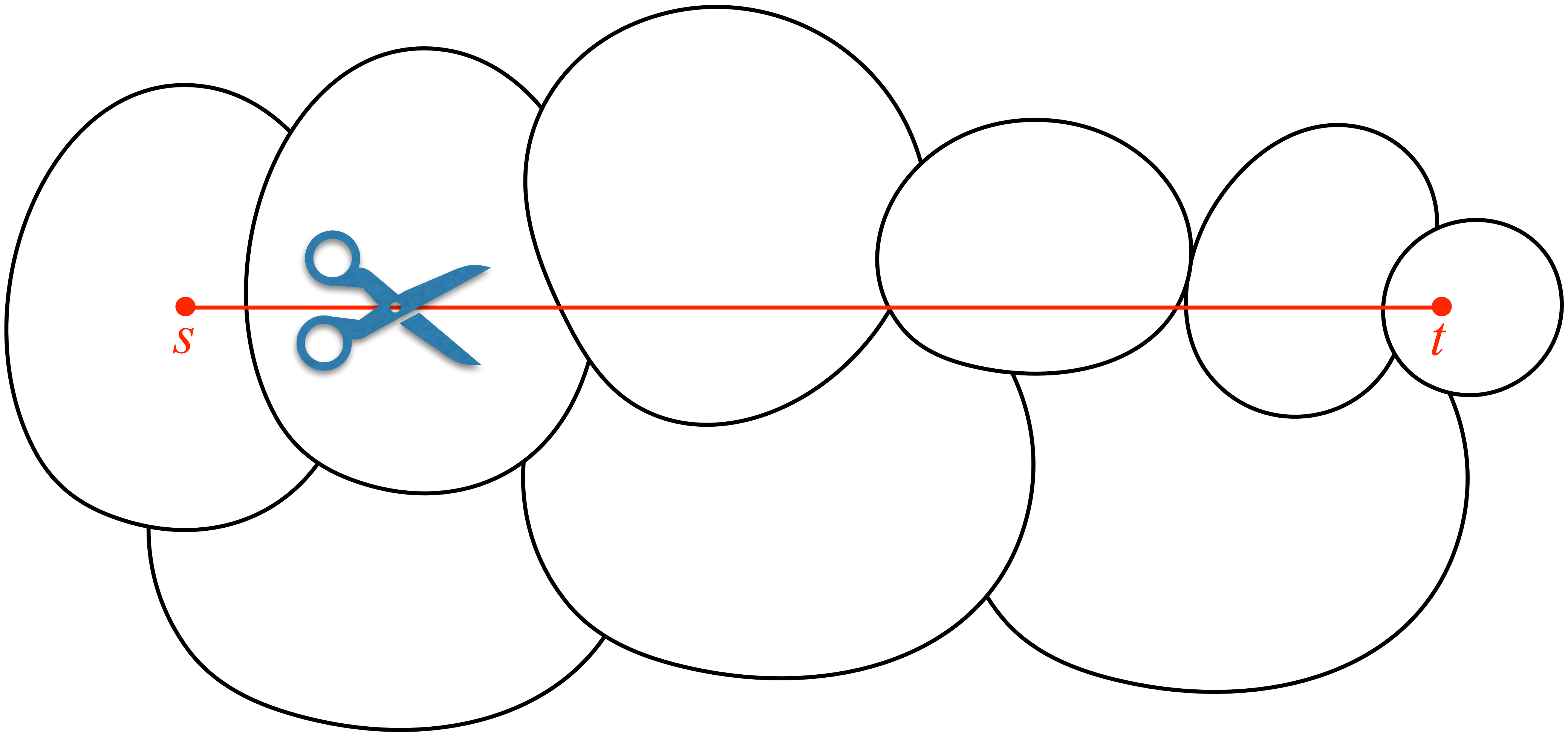
[Łącki, Sankowski ESA 2011]



Bottleneck: recomputing DDG in each recursive level (O(nloglogn) time)

Undirected Min cycle in O(nloglogn) time

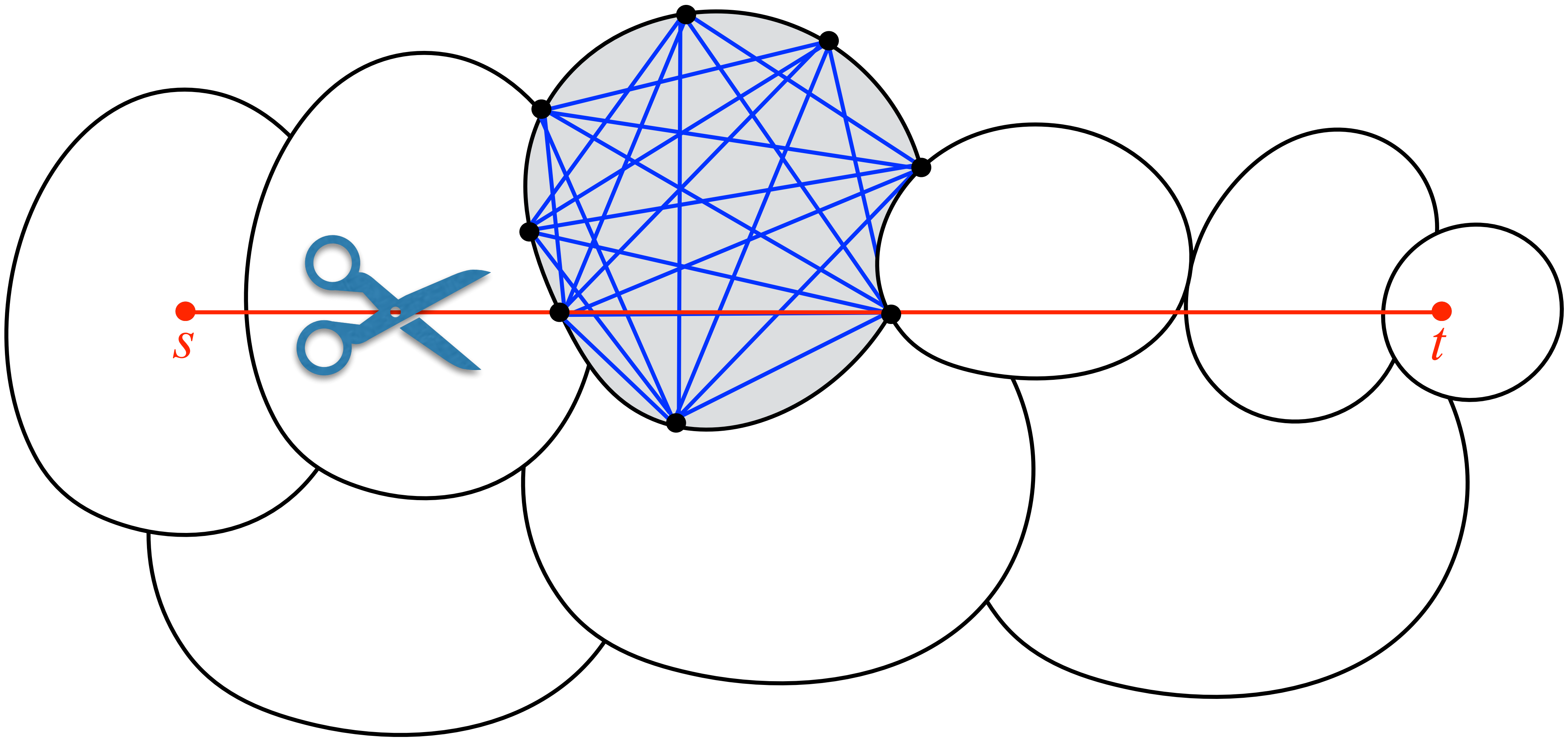
[Łącki, Sankowski ESA 2011]



Bottleneck: recomputing DDG in each recursive level (O(nloglogn) time)

Undirected Min cycle in O(nloglogn) time

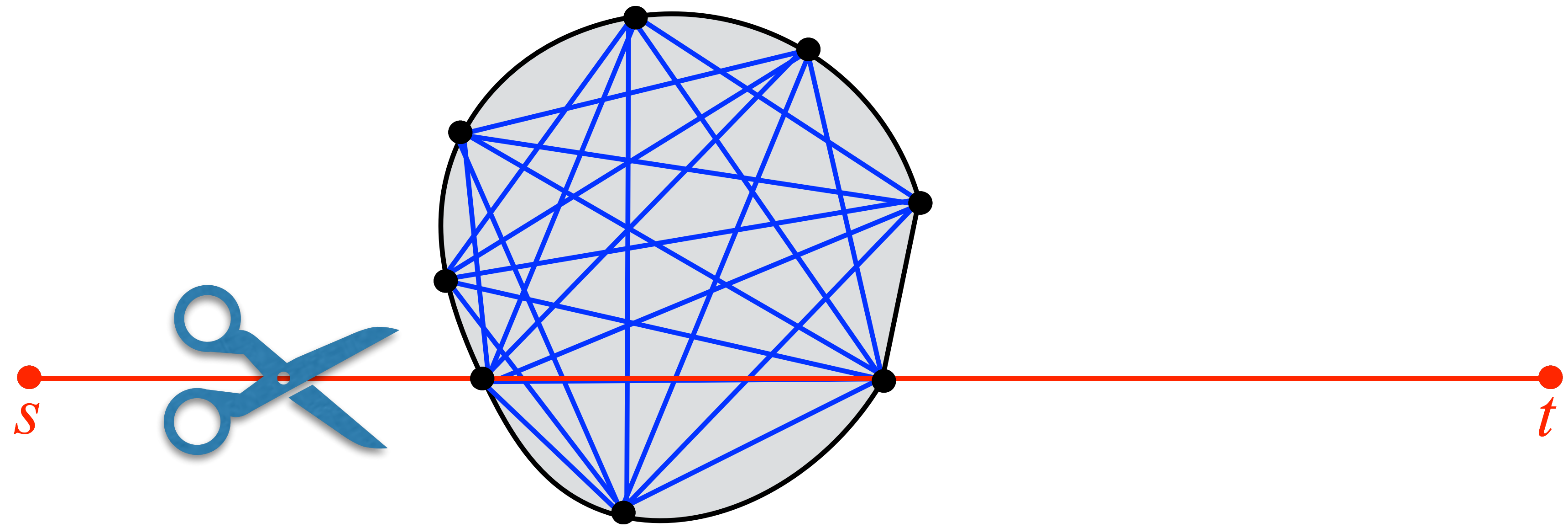
[Łącki, Sankowski ESA 2011]



Bottleneck: recomputing DDG in each recursive level (O(nloglogn) time)

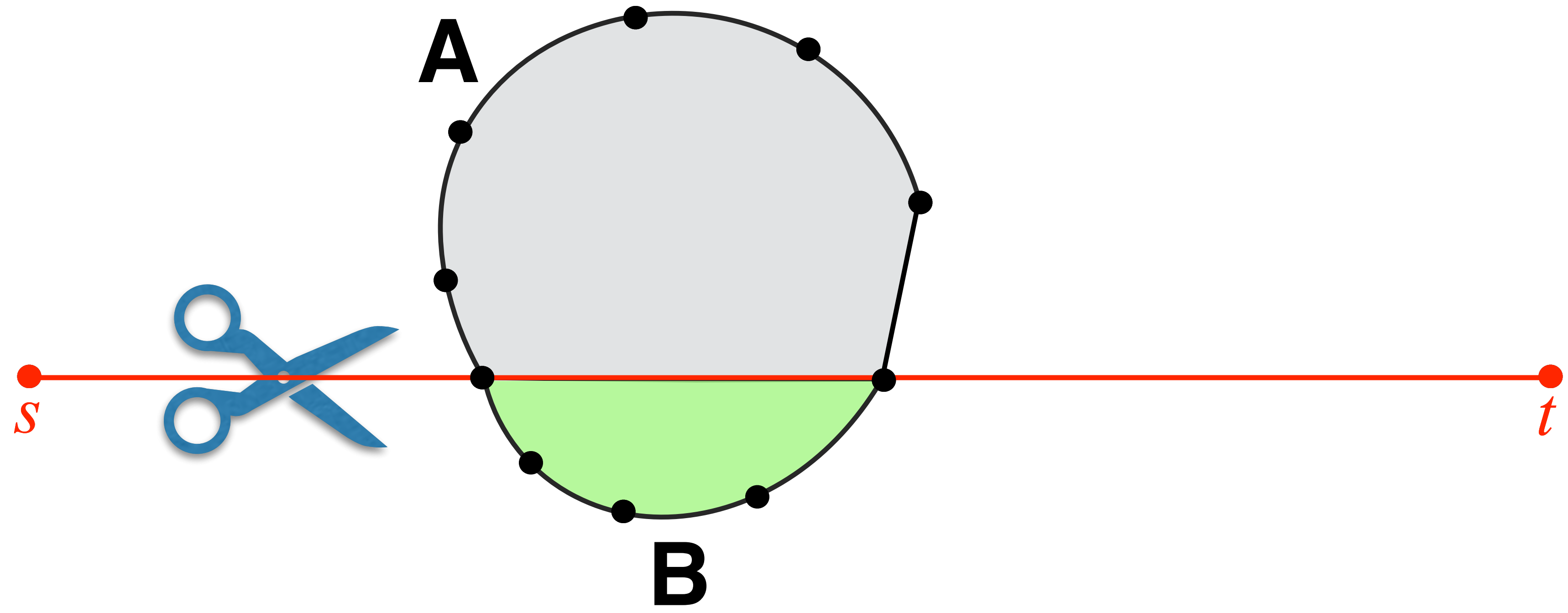
Undirected Min cycle in O(nloglogn) time

[Łącki, Sankowski ESA 2011]



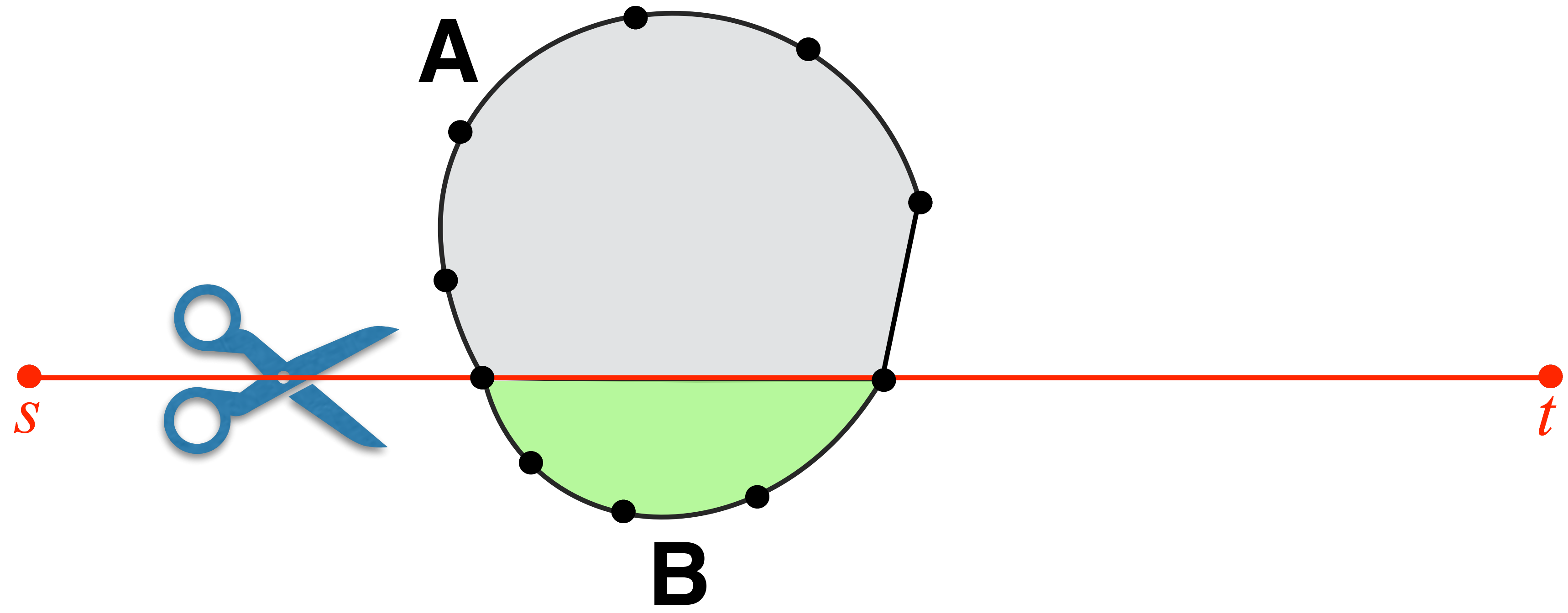
Undirected Min cycle in O(nloglogn) time

[Łącki, Sankowski ESA 2011]

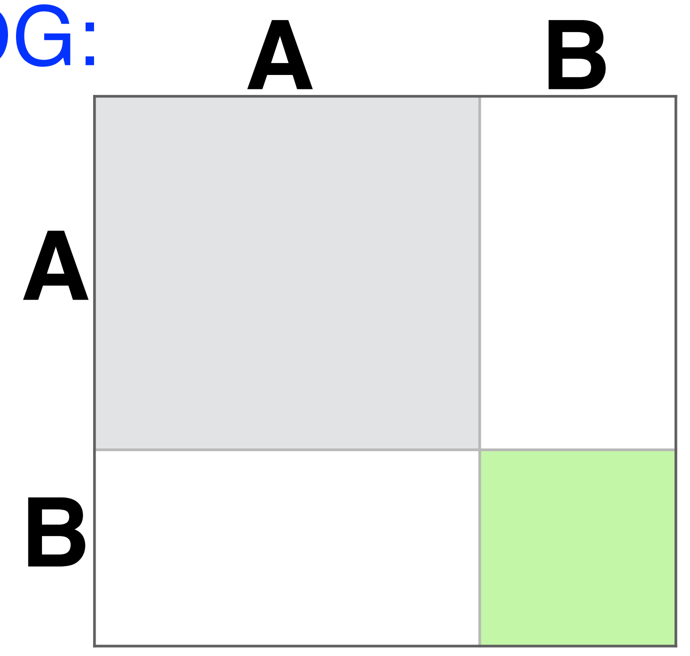


Undirected Min cycle in O(nloglogn) time

[Łącki, Sankowski ESA 2011]

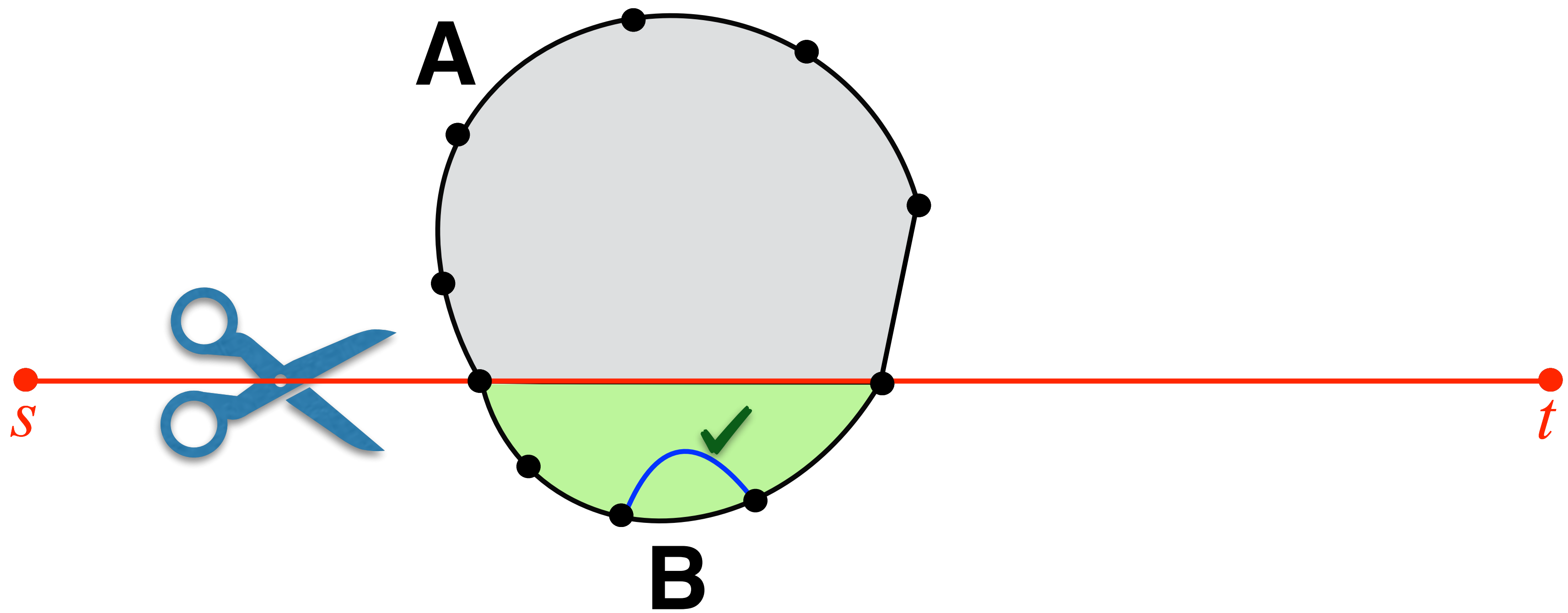


DDG:

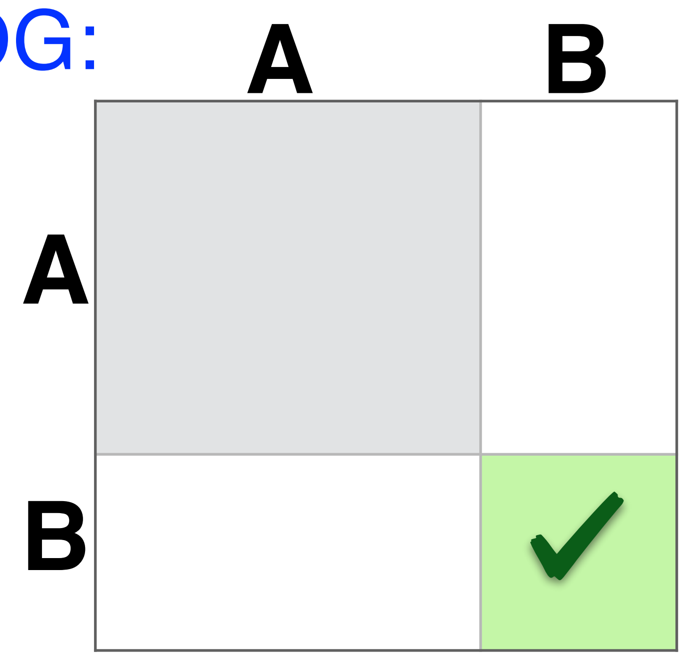


Undirected Min cycle in O(nloglogn) time

[Łącki, Sankowski ESA 2011]

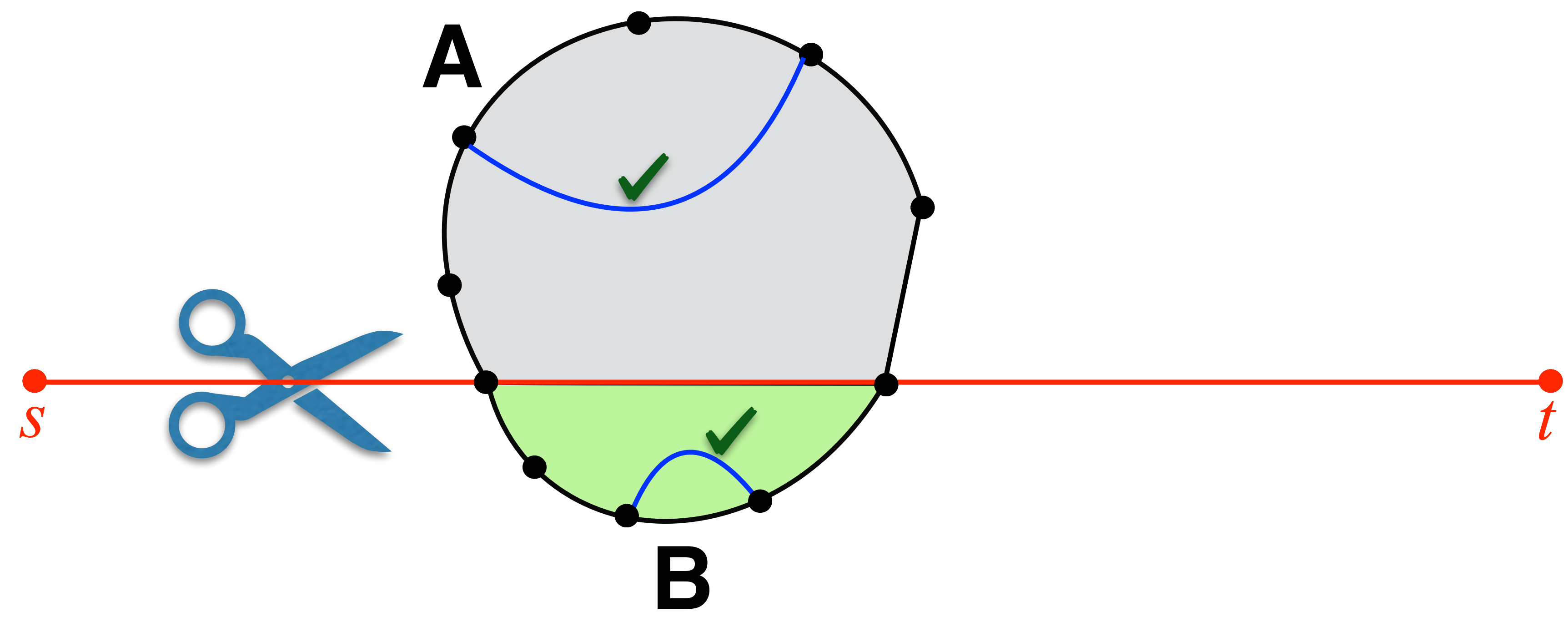


DDG:



Undirected Min cycle in O(nloglogn) time

[Łącki, Sankowski ESA 2011]

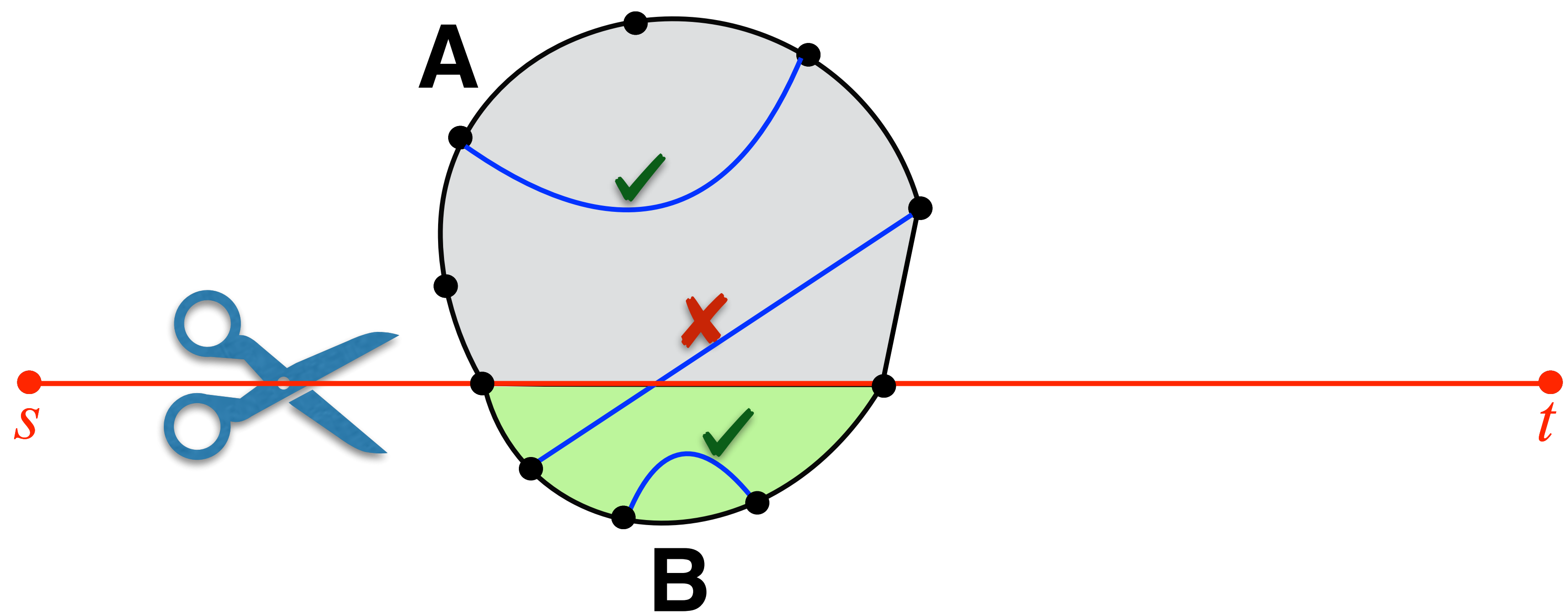


DDG:

	A	B
A	✓	
B		✓

Undirected Min cycle in O(nloglogn) time

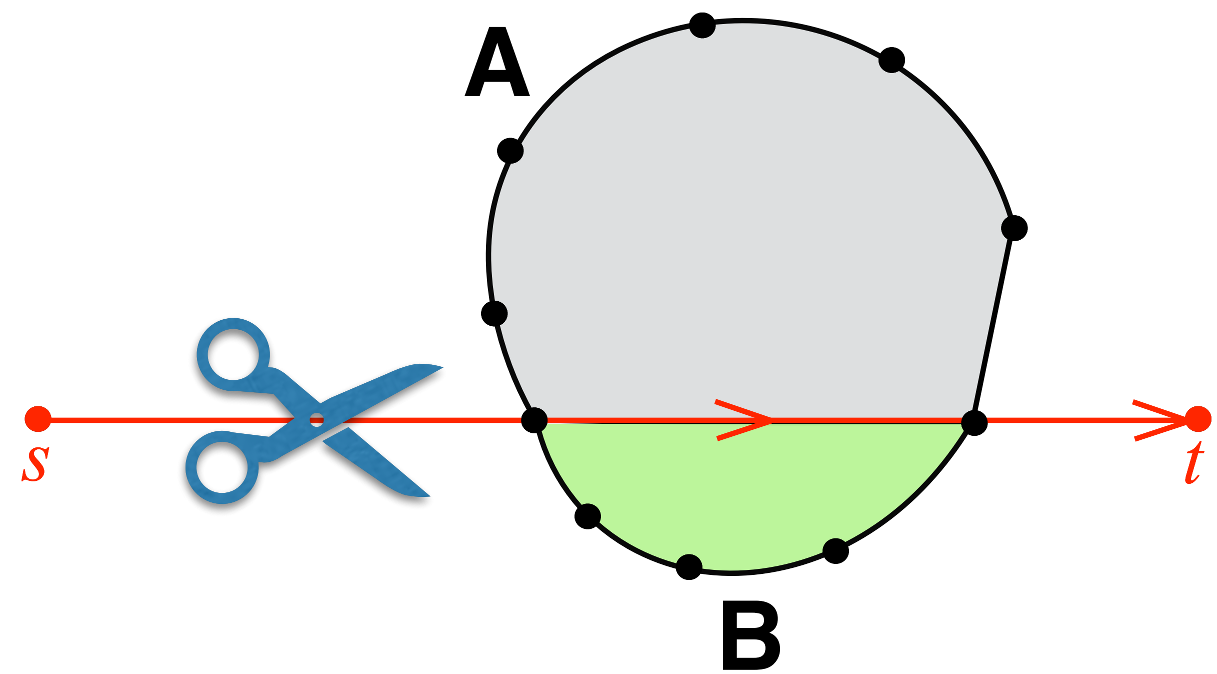
[Łącki, Sankowski ESA 2011]



DDG:

	A	B
A	✓	✗
B	✗	✓

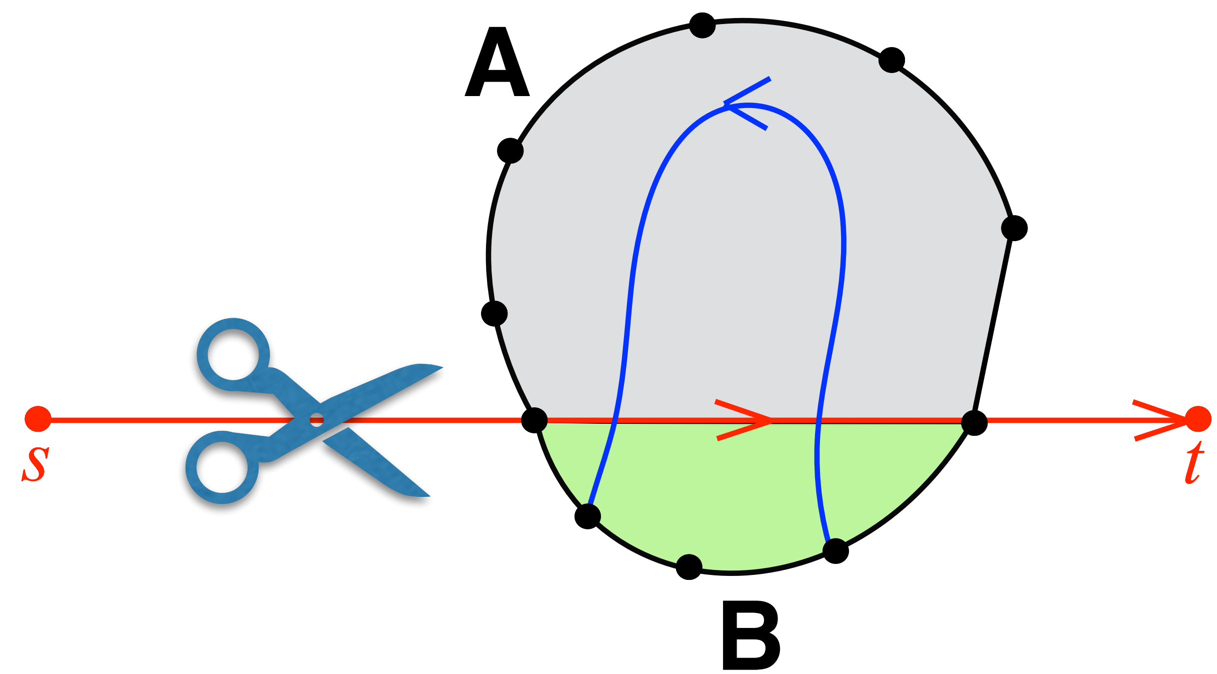
Directed Min cycle in O(nloglogn) time

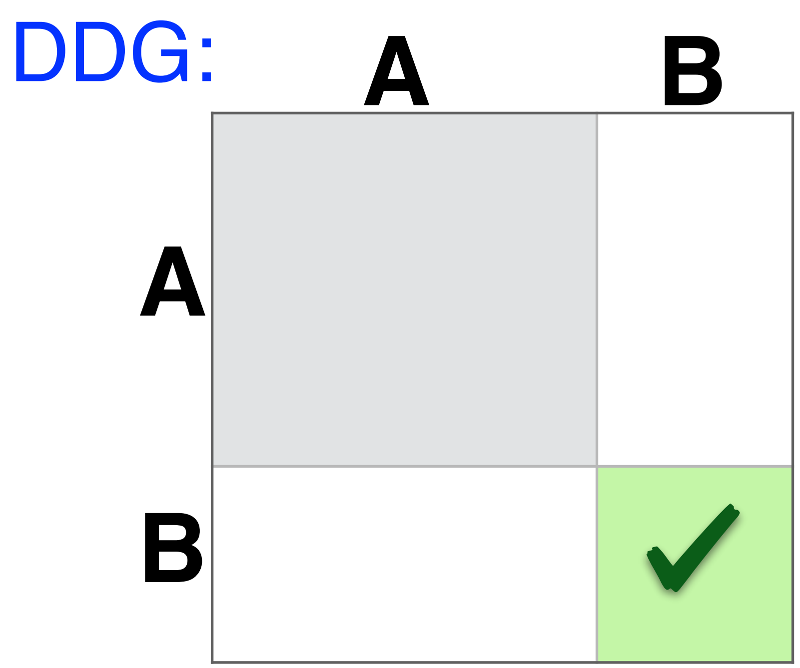


DDG:

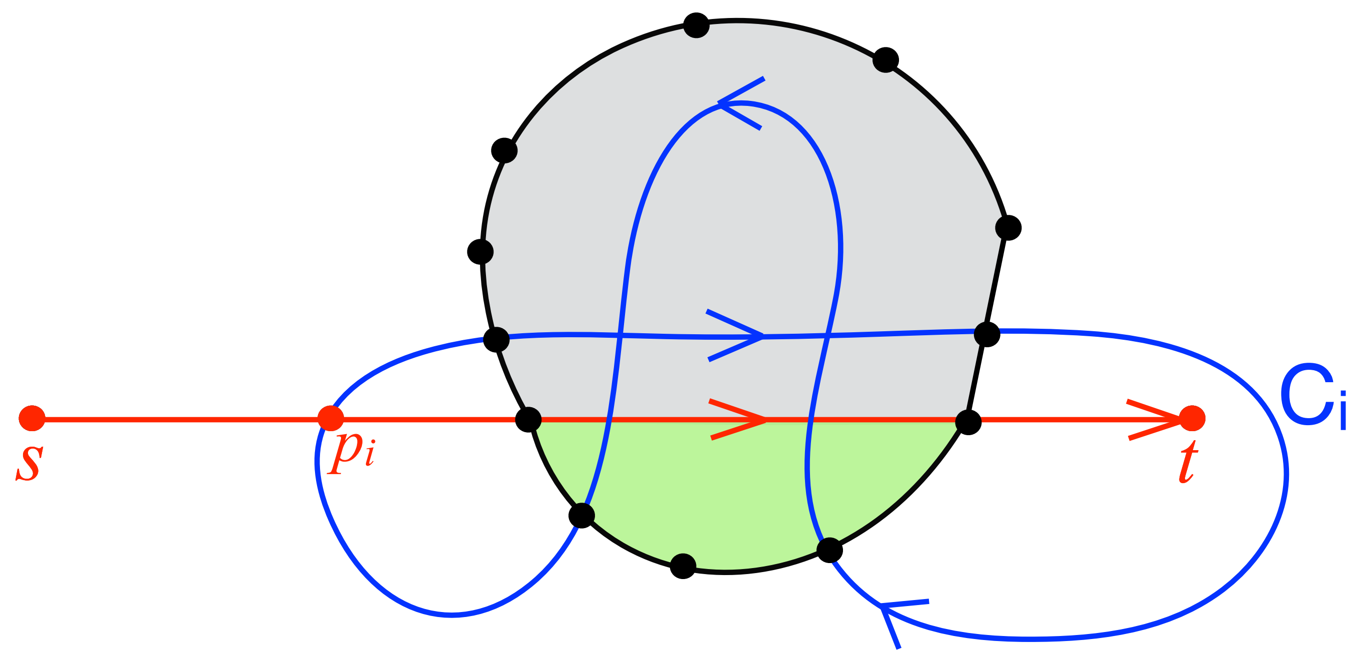
	A	B
A	✓	✗
B	✗	✓

Directed Min cycle in O(nloglogn) time



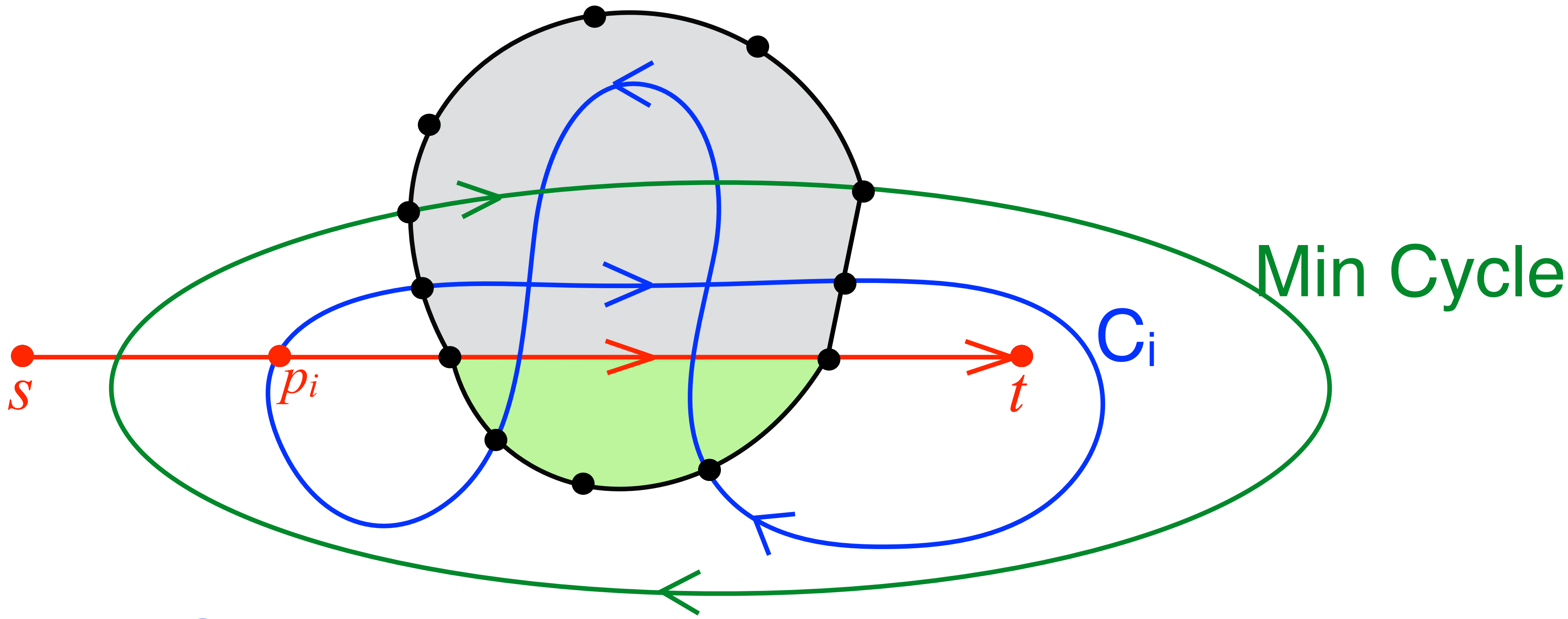


Directed Min cycle in O(nloglogn) time



The underlying C_i might be non-simple!

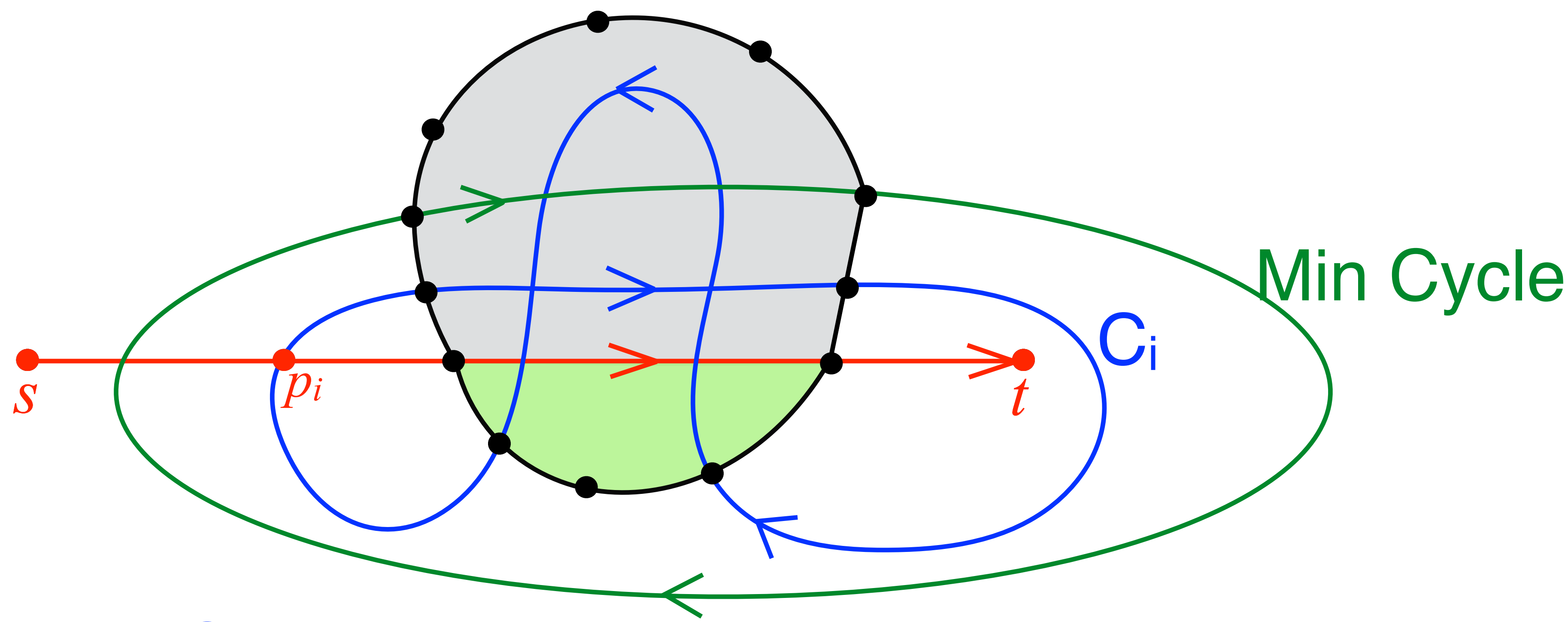
Directed Min cycle in O(nloglogn) time



The underlying C_i might be non-simple!

The underlying Min Cycle might cross C_i!

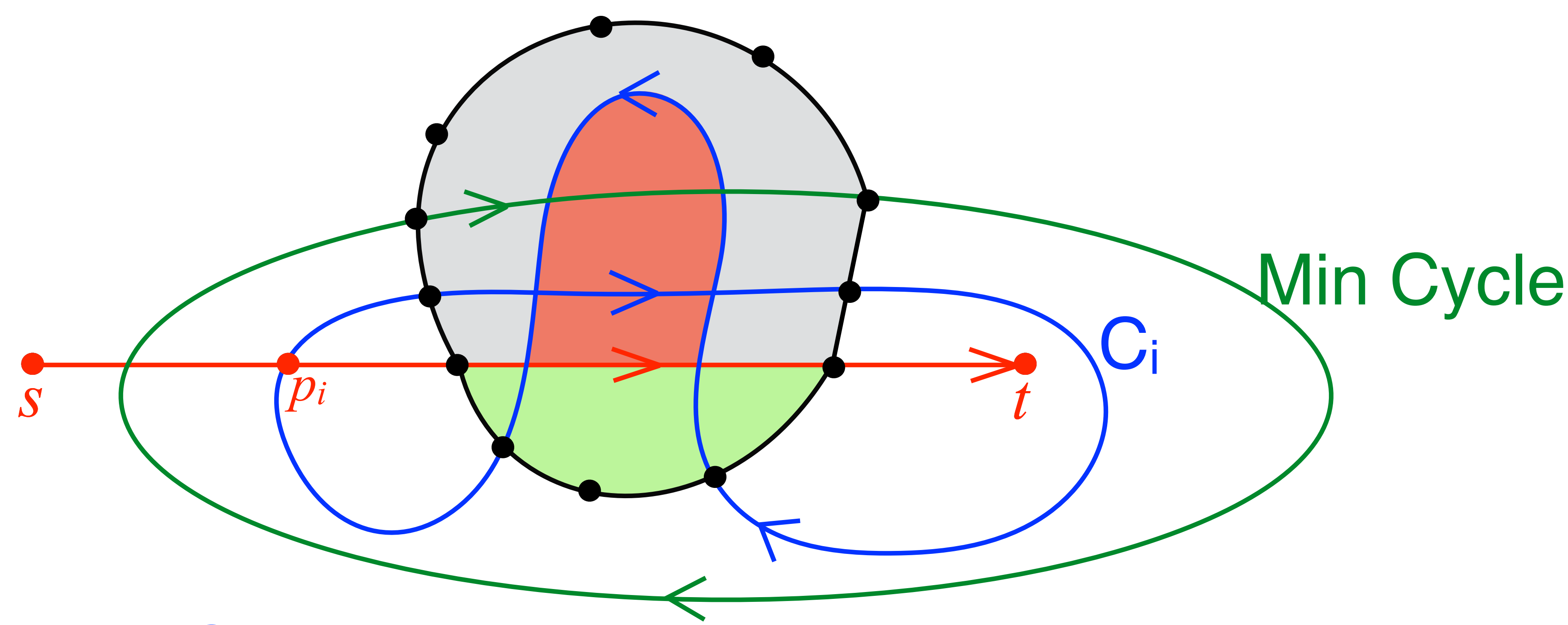
Directed Min cycle in O(nloglogn) time



The underlying C_i might be non-simple!
The underlying Min Cycle might cross C_i!

Theorem:
We do not loose the Min Cycle after cutting the DDG along C_i

Directed Min cycle in O(nloglogn) time



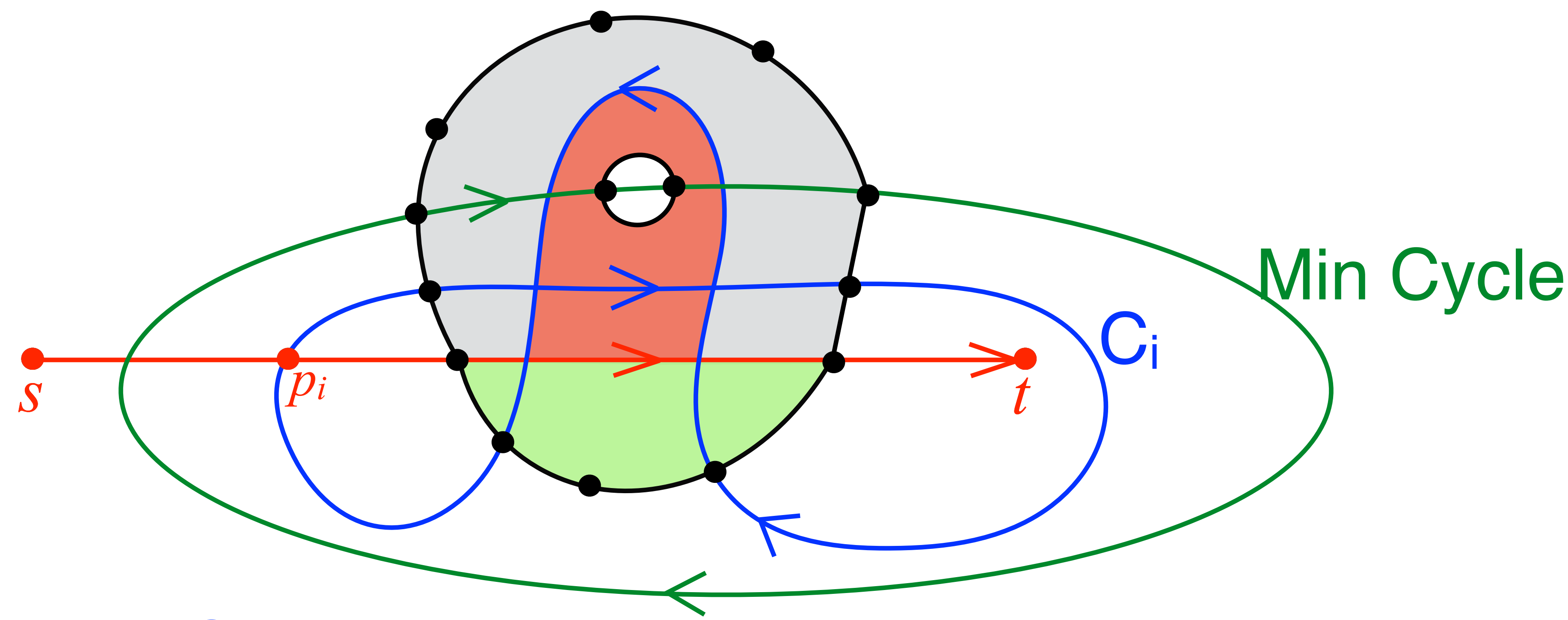
The underlying C_i might be non-simple!

The underlying Min Cycle might cross C_i!

Theorem:

We do not loose the Min Cycle after cutting the DDG along C_i as long as there are **no holes trapped in the crossing area**

Directed Min cycle in O(nloglogn) time



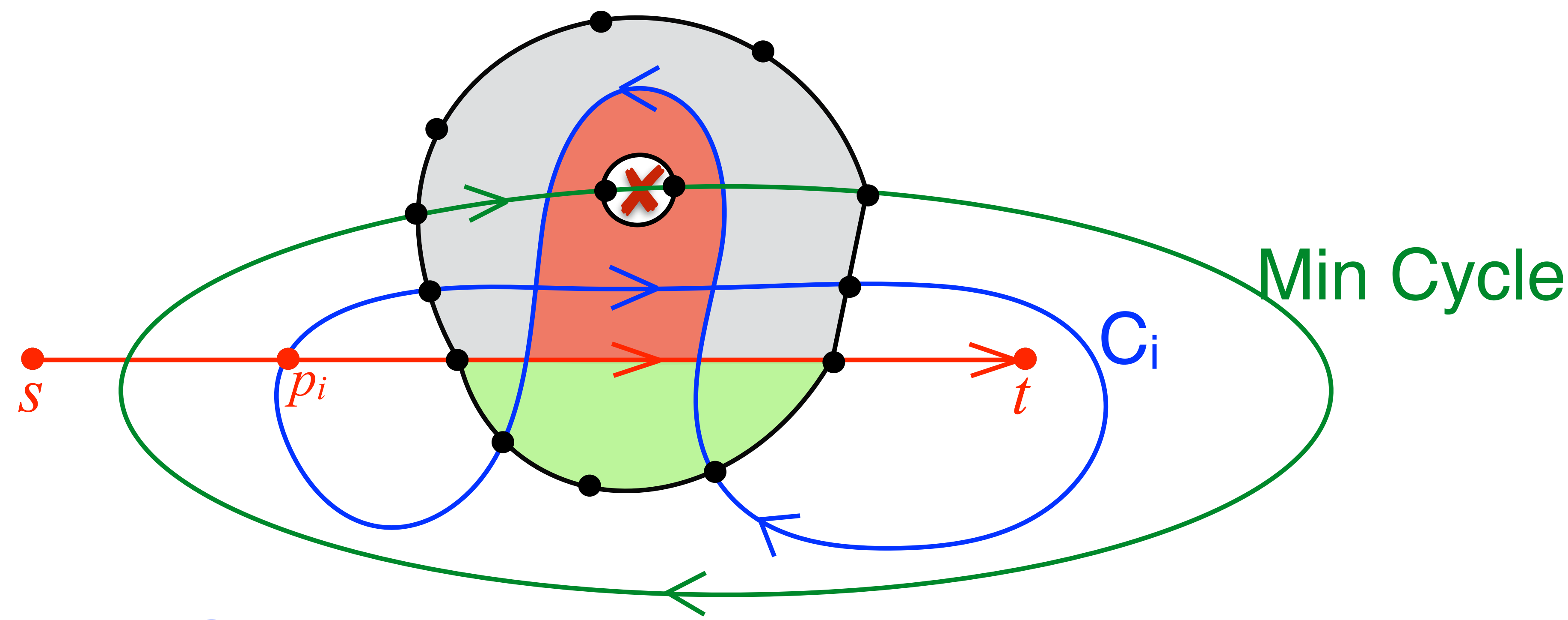
The underlying C_i might be non-simple!

The underlying Min Cycle might cross C_i!

Theorem:

We do not loose the Min Cycle after cutting the DDG along C_i as long as there are **no holes trapped in the crossing area**

Directed Min cycle in O(nloglogn) time



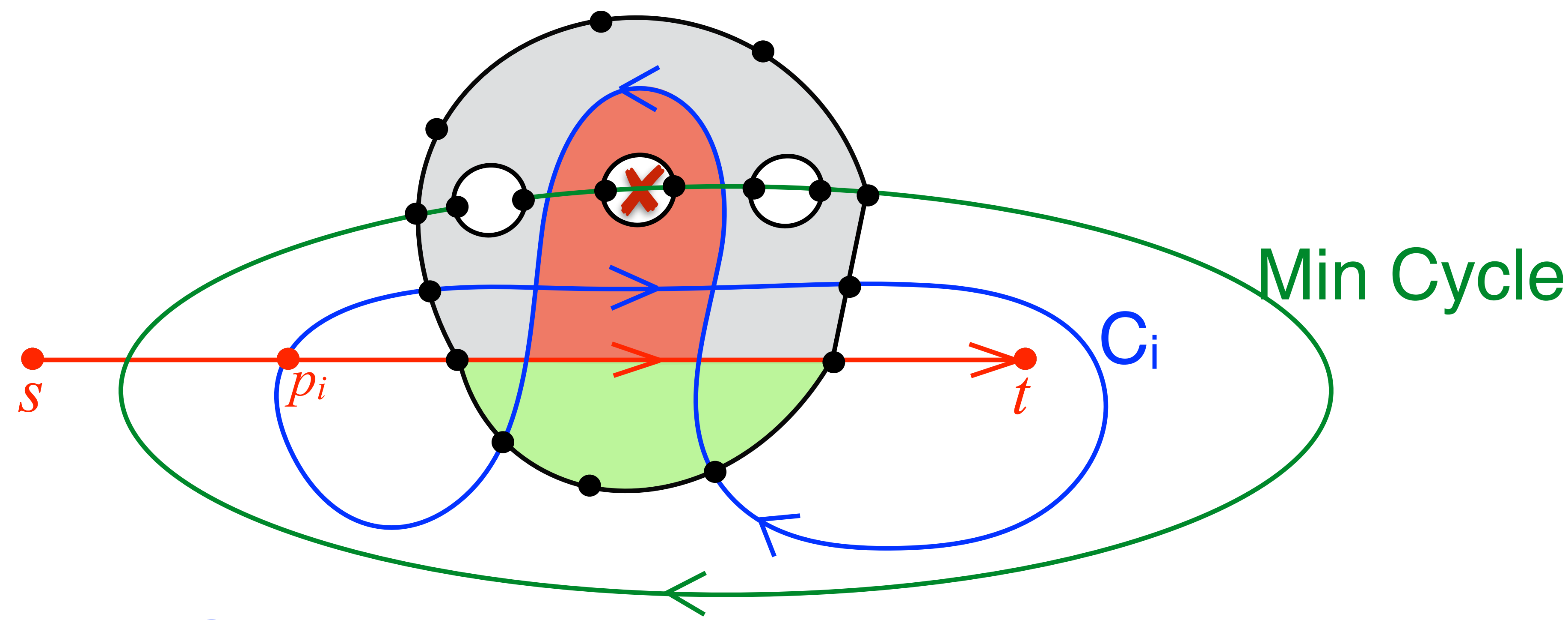
The underlying C_i might be non-simple!

The underlying Min Cycle might cross C_i!

Theorem:

We do not loose the Min Cycle after cutting the DDG along C_i as long as there are **no holes trapped in the crossing area**

Directed Min cycle in O(nloglogn) time



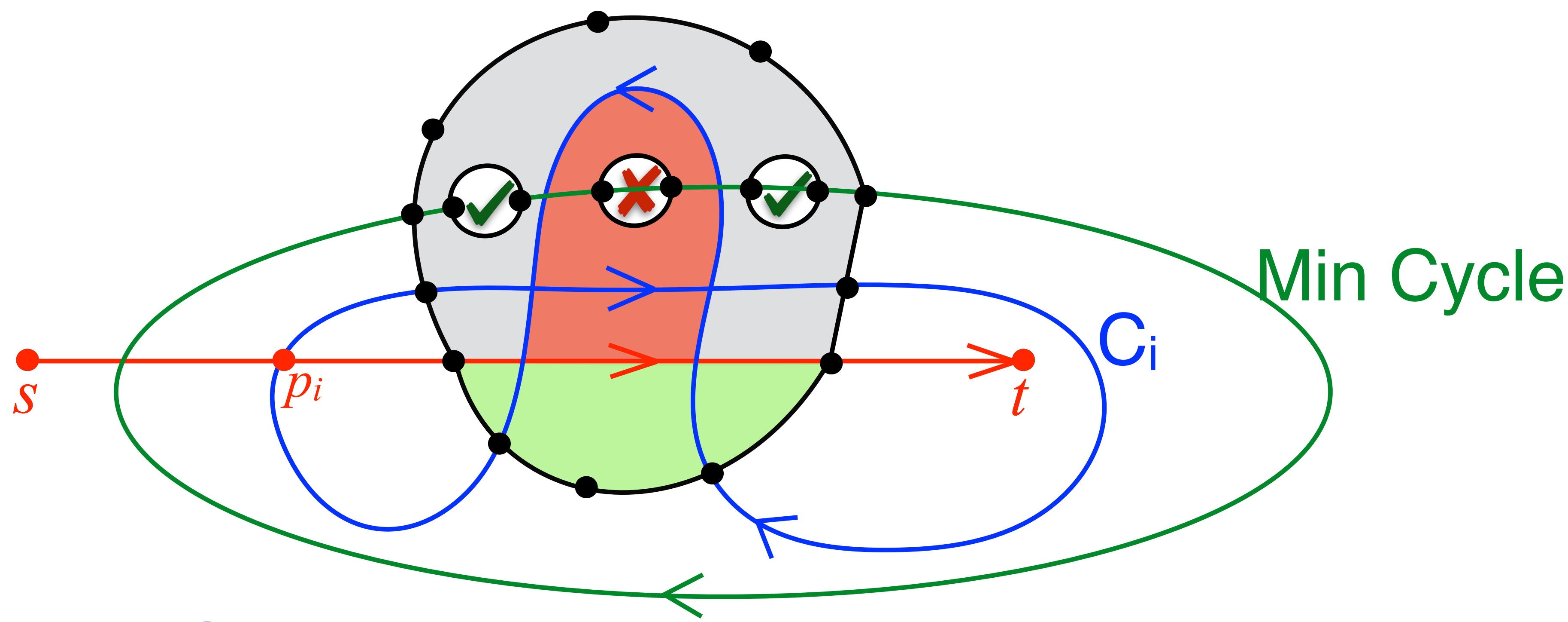
The underlying C_i might be non-simple!

The underlying Min Cycle might cross C_i!

Theorem:

We do not loose the Min Cycle after cutting the DDG along C_i as long as there are **no holes trapped in the crossing area**

<u>Directed</u> Min cycle in O(nloglogn) time



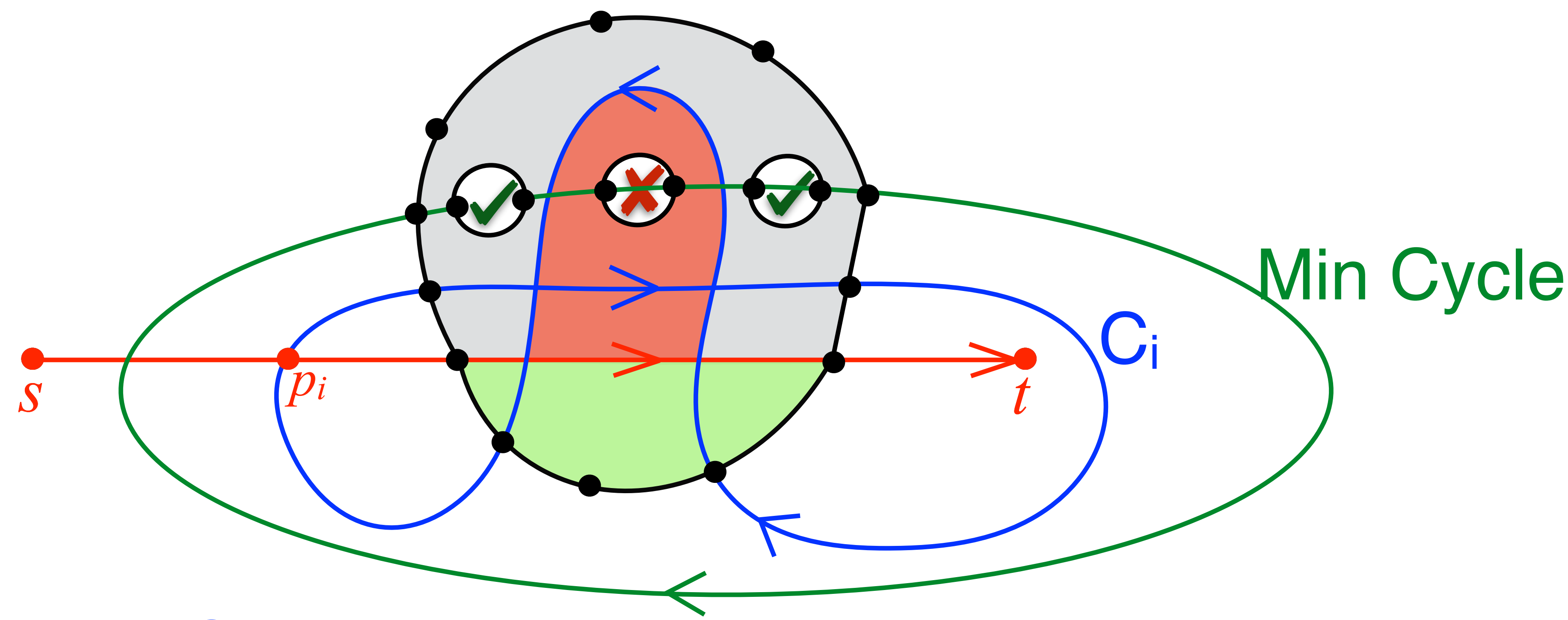
The underlying C_i might be non-simple!

The underlying Min Cycle might cross C_i!

<u>Theorem</u>:

We do not loose the Min Cycle after cutting the DDG along C_i as long as there are **no holes trapped in the crossing area**

Directed Min cycle in O(nloglogn) time



The underlying C_i might be non-simple!

The underlying Min Cycle might cross C_i!

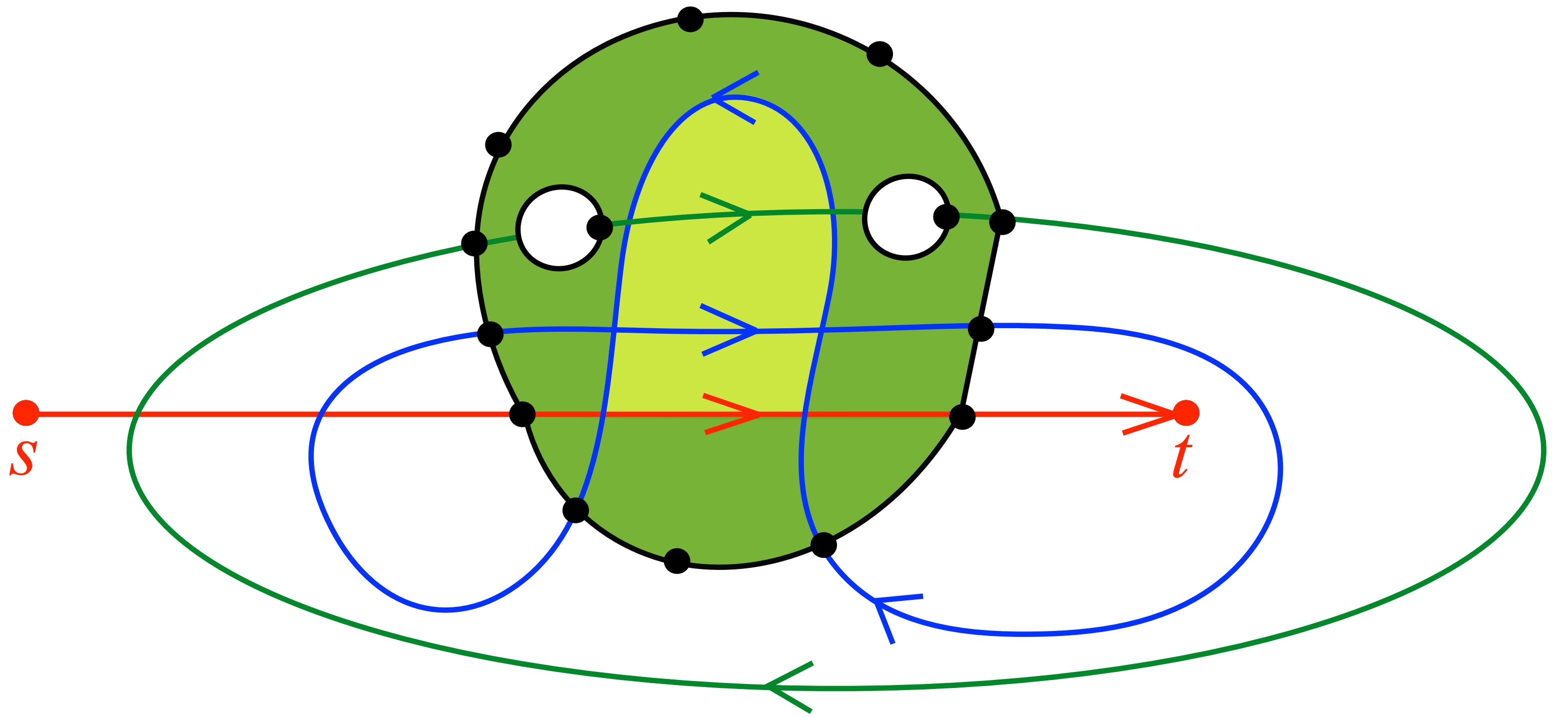
Theorem:

We do not loose the Min Cycle after cutting the DDG along C_i as long as there are **no holes trapped in the crossing area**

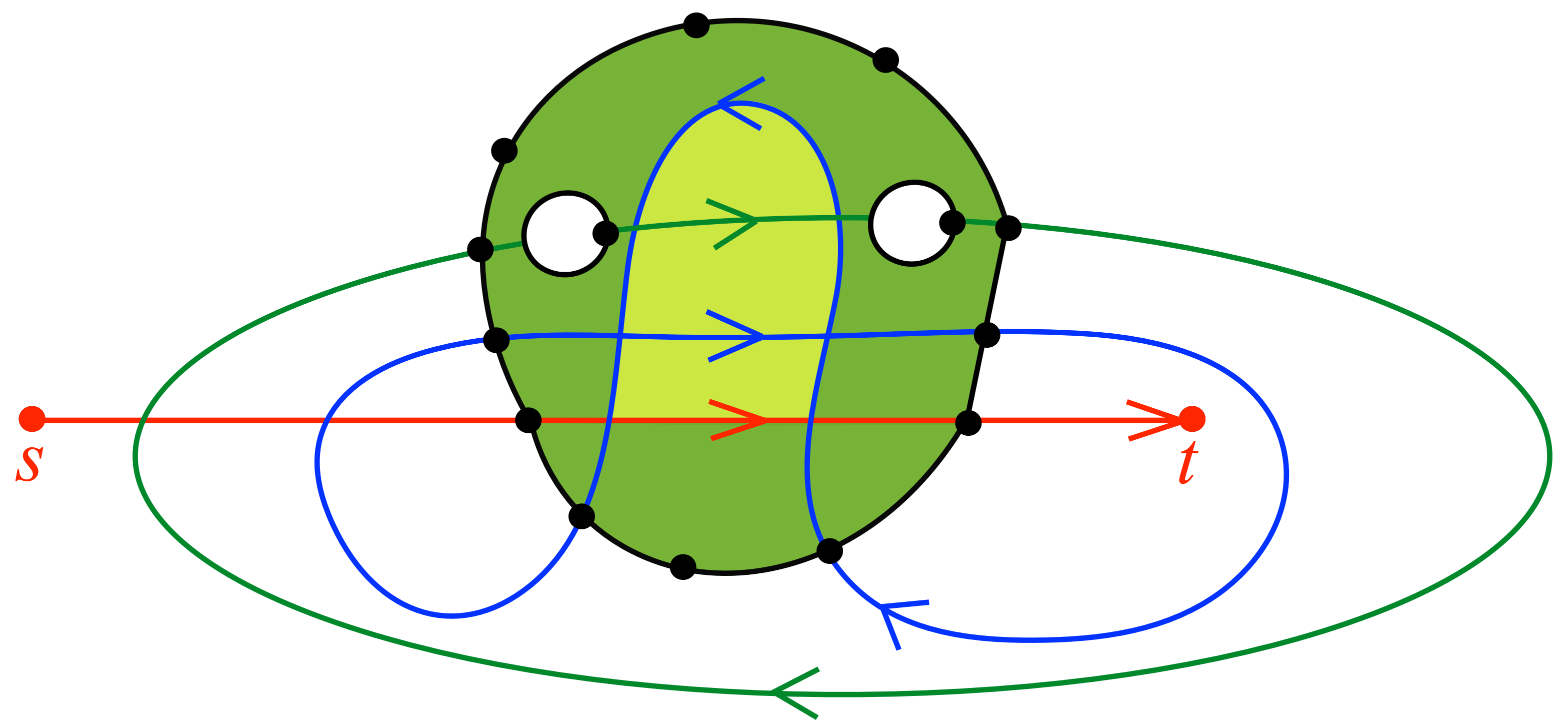
Challenges:

1. prove this theorem.
2. prevent holes in crossing area (by building one DDG for every hole configuration, using a bounded genus graph)

Conclusions



Conclusions



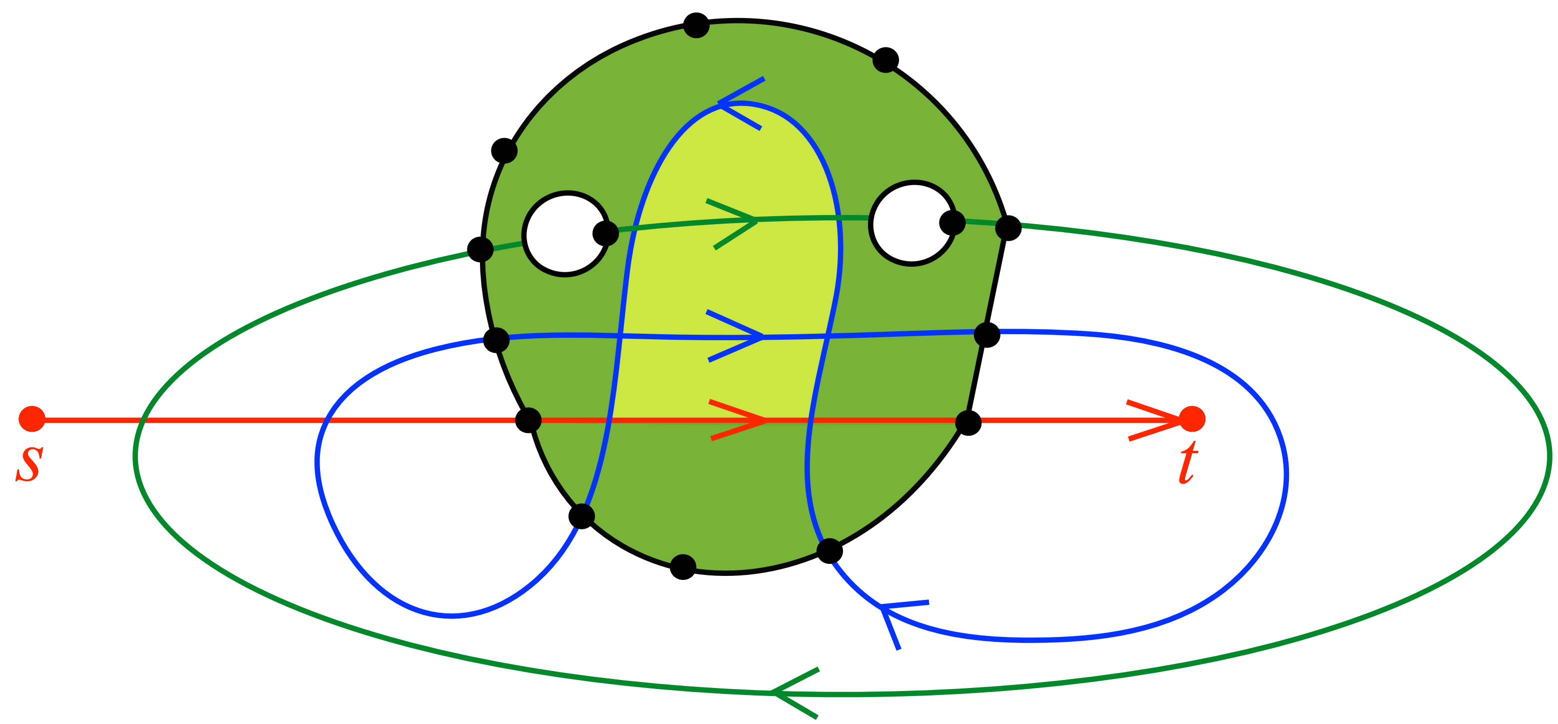
Undirected min st-cut: O(n loglogn)
[Italiano, Nussbaum, Sankowski, Wulff-Nilsen STOC 2011]

Undirected min cut: O(n loglogn)
[Łącki, Sankowski ESA 2011]

Directed min cut O(n loglogn)

Directed min st-cut: O(n logn)
[Borradaile, Klein SODA 2006]

Conclusions



Undirected min st-cut: O(n loglogn)
[Italiano, Nussbaum, Sankowski, Wulff-Nilsen STOC 2011]

Undirected min cut: O(n loglogn)
[Łącki, Sankowski ESA 2011]

Directed min cut O(n loglogn)

Directed min st-cut: O(n logn)
[Borradaile, Klein SODA 2006]

Thank Yu