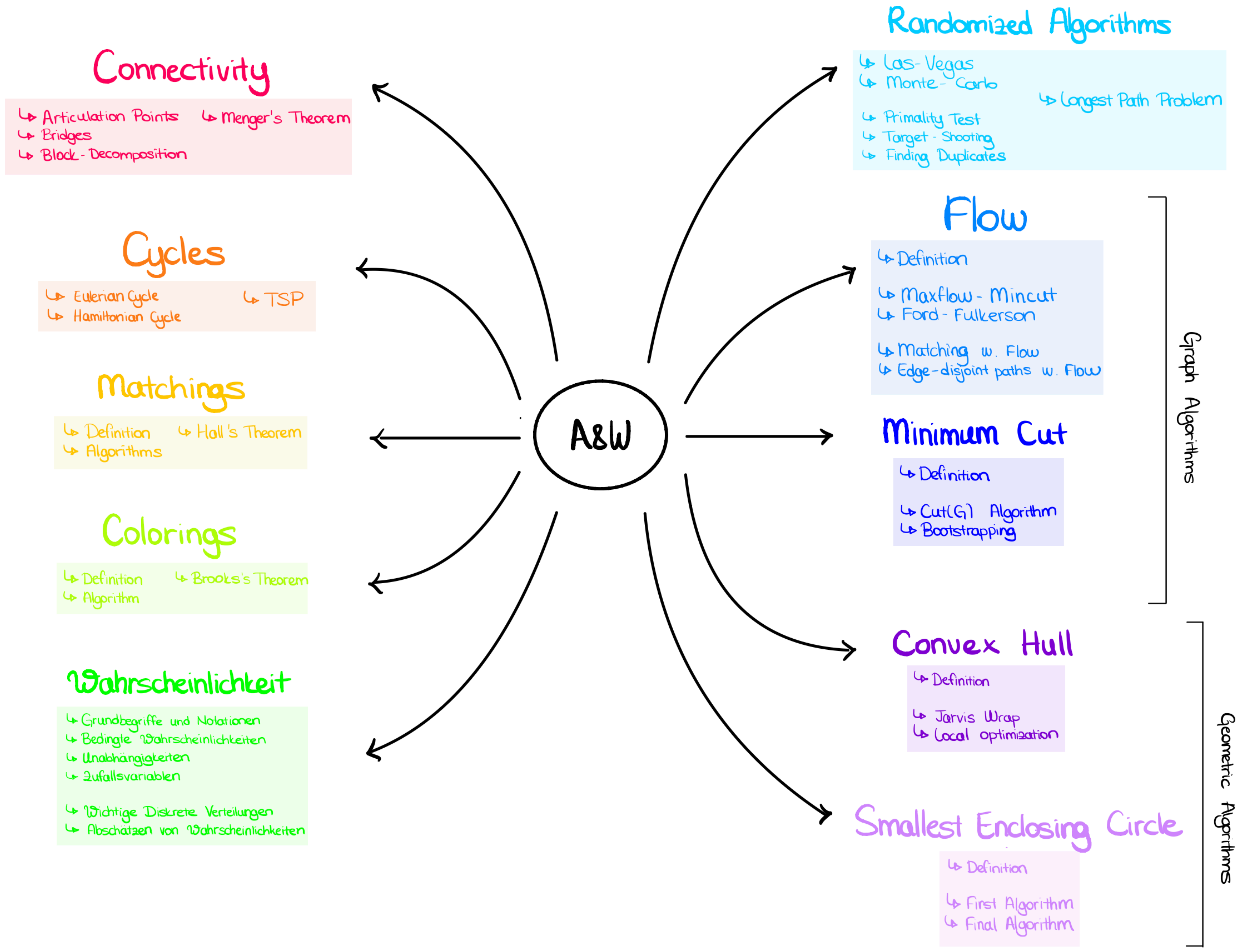
A&W

Exercise Session 12

Minimum Cut, Smallest Enclosing Cycle

Nil Ozer

A&W Overview



A&W
Connectivity
Articulation Points
Bridges
Block - Decomposition
Menger's Theorem
Cycles
Eulerian Cycle
Hamiltonian Cycle
TSP
Matchings
Definition
Algorithms
Hall's Theorem
Colorings
Definition
Algorithm
Brooks's Theorem
Wahrscheinlichkeit
Grundbegriffe und Notationen
Bedingte Wahrscheinlichkeiten
Unabhängigkeiten
Zufallsvariablen
Wichtige Diskrete Verteilungen
Abschätzen von Wahrscheinlichkeiten
Randomized Algorithms
Las-Vegas
Monte-Carlo
Primality Test
Target-Shooting
Finding Duplicates
Longest Path Problem
Flow
Definition
Maxflow - Mincut
Ford - Fulkerson
Matching w. Flow
Edge-disjoint paths w. Flow
Minimum Cut
Definition
Cut(G) Algorithm
Bootstrapping
Graph Algorithms
Convex Hull
Definition
Jarvis Wrap
Local optimization
Smallest Enclosing Circle
Definition
First Algorithm
Final Algorithm
Geometric Algorithms

Last Weeks …

- 08.05 : Randomized Algorithms II
- 15.05 : Flow
- 23.05 online : Minimum Cut , Smallest Enclosing Circle
- 28.05 extra session : Exam Prep Session + Pizza and Drinks
- 30.05 last extra session : Convex Hull (shortly remaining primality tests)

Outline

- Minimum Cut
- Smallest Enclosing Circle

Minimum Cut

Min-Cut

Definitions

- Multigraph :
  - undirected, unweighted, without self-loops
  - possibly with multiple edges between the same pair of nodes

Min-Cut

Definitions

- Multigraph :
  - undirected, unweighted, without self-loops
  - possibly with multiple edges between the same pair of nodes

- Edge Cut C :
  - A set of edges C s.t. $G' = (V, E \backslash C)$ is a disconnected graph.

Min-Cut
Definitions

- Multigraph :
  - undirected, unweighted, without self-loops
  - possibly with multiple edges between the same pair of nodes
- Edge Cut C :
  - A set of edges C s.t. $G' = (V, E \backslash C)$ is a disconnected graph.
- $\mu(G)$:
  - the cardinality of the smallest possible edge cut in graph G.

$$\mu(G) := \min_{\substack{C \subseteq E, \\ (V, E \backslash C) \text{ disconnected}}} |C|$$

Min-Cut

Problem Description

given : A multigraph G

to find : $\mu(G)$

- Multigraph :
  - undirected, unweighted, without self-loops
  - possibly with multiple edges between the same pair of nodes
- Edge Cut C :
  - A set of edges C s.t. $G' = (V, E \backslash C)$ is a disconnected graph.
- $\mu(G)$:
  - the cardinality of the smallest possible edge cut in graph G.

$$\mu(G) := \min_{\substack{C \subseteq E, \\ (V, E\backslash C) \text{ disconnected}}} |C|$$

Min-Cut

Problem Description

given : A multigraph G

to find : $\mu(G)$

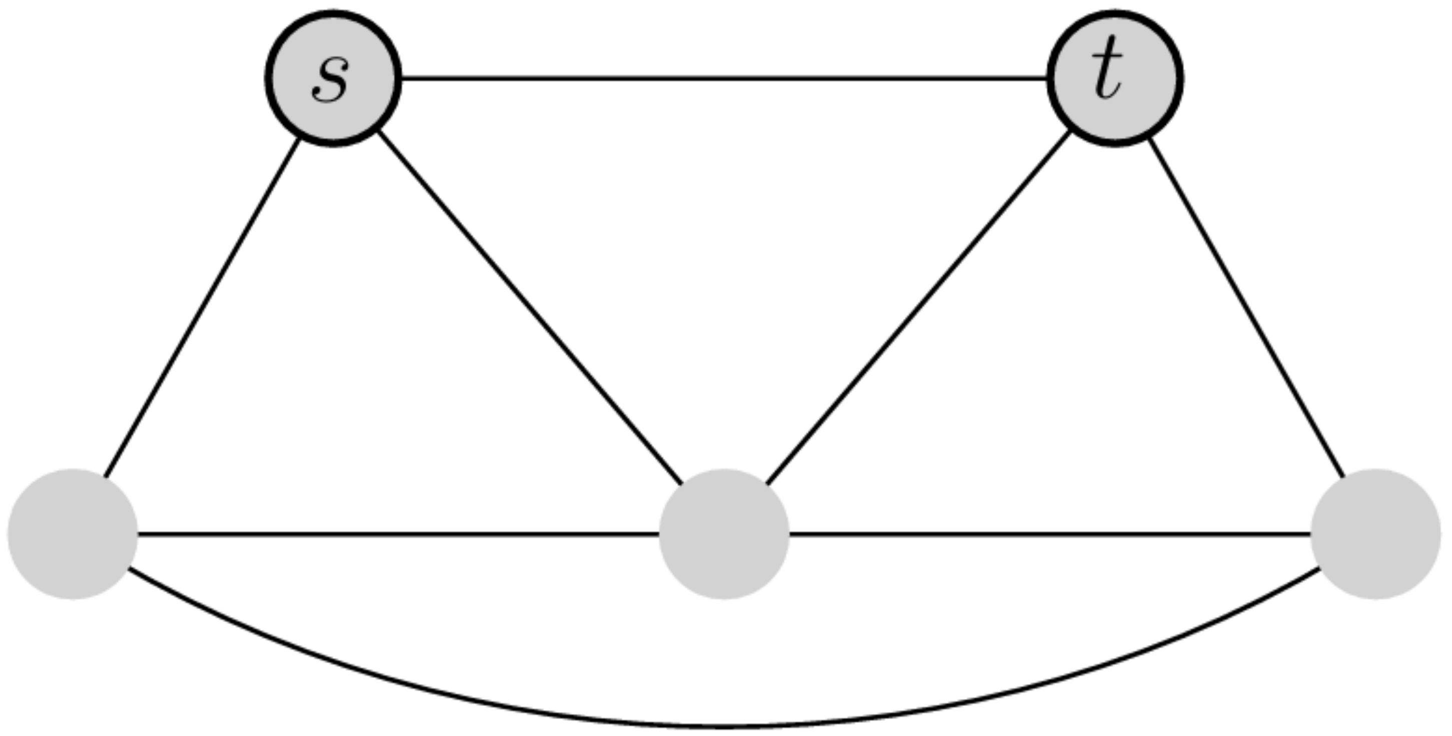
Examples :

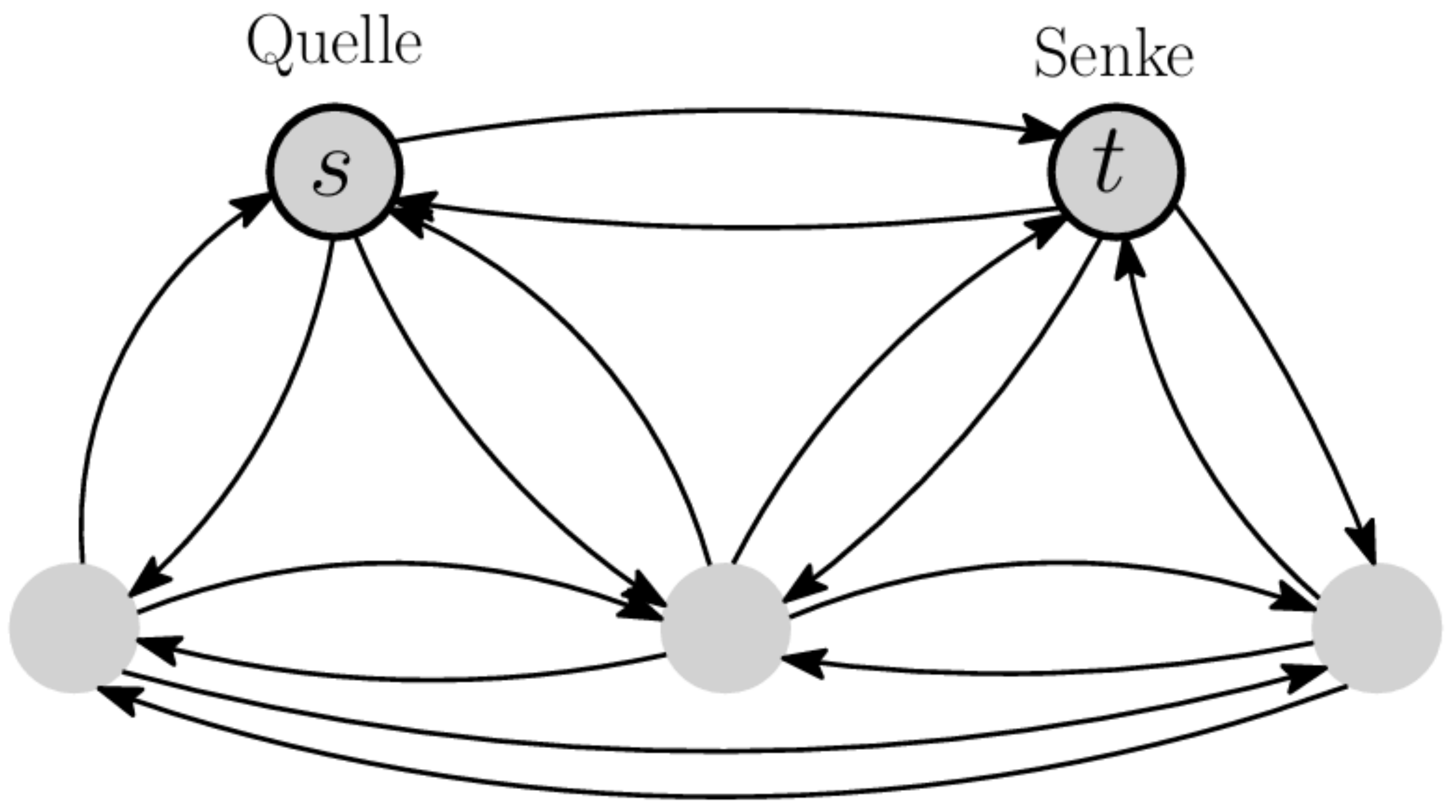
- Multigraph :
  - undirected, unweighted, without self-loops
  - possibly with multiple edges between the same pair of nodes
- Edge Cut C :
  - A set of edges C s.t. $G' = (V, E \backslash C)$ is a disconnected graph.
- $\mu(G)$:
  - the cardinality of the smallest possible edge cut in graph G.

$$\mu(G) := \min_{\substack{C \subseteq E, \\ (V, E \backslash C) \text{ disconnected}}} |C|$$

Min-Cut
First Known Solution

minimum s-t cut in O(mn logn)





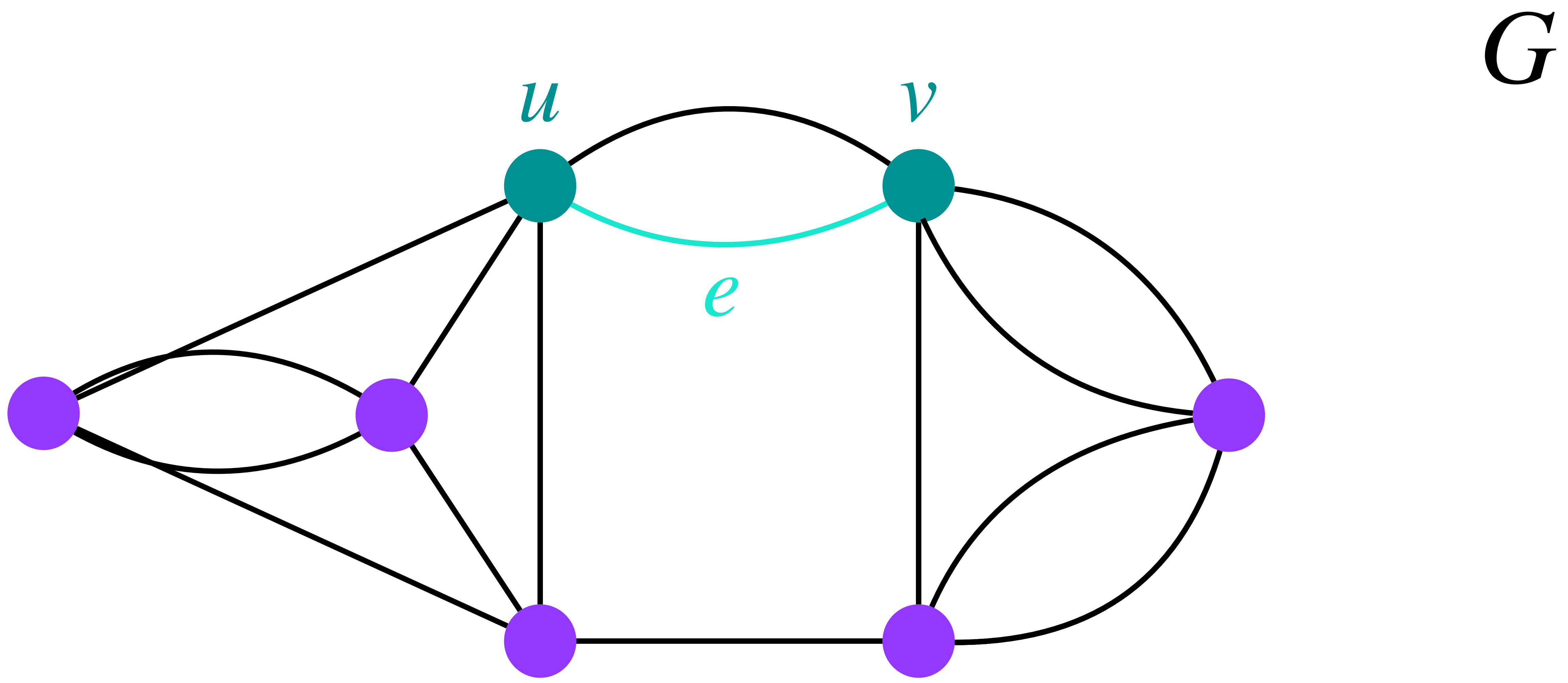
- Fix a source node s
- Then consider $t \in V \setminus \{s\}$, for each t compute the minimum s-t cut
- The global min-cut is the smallest of these s-t cuts

total runtime : $O((n-1)mn \log n) = O(n^4 \log n)$

Min-Cut

Edge Contraction of e

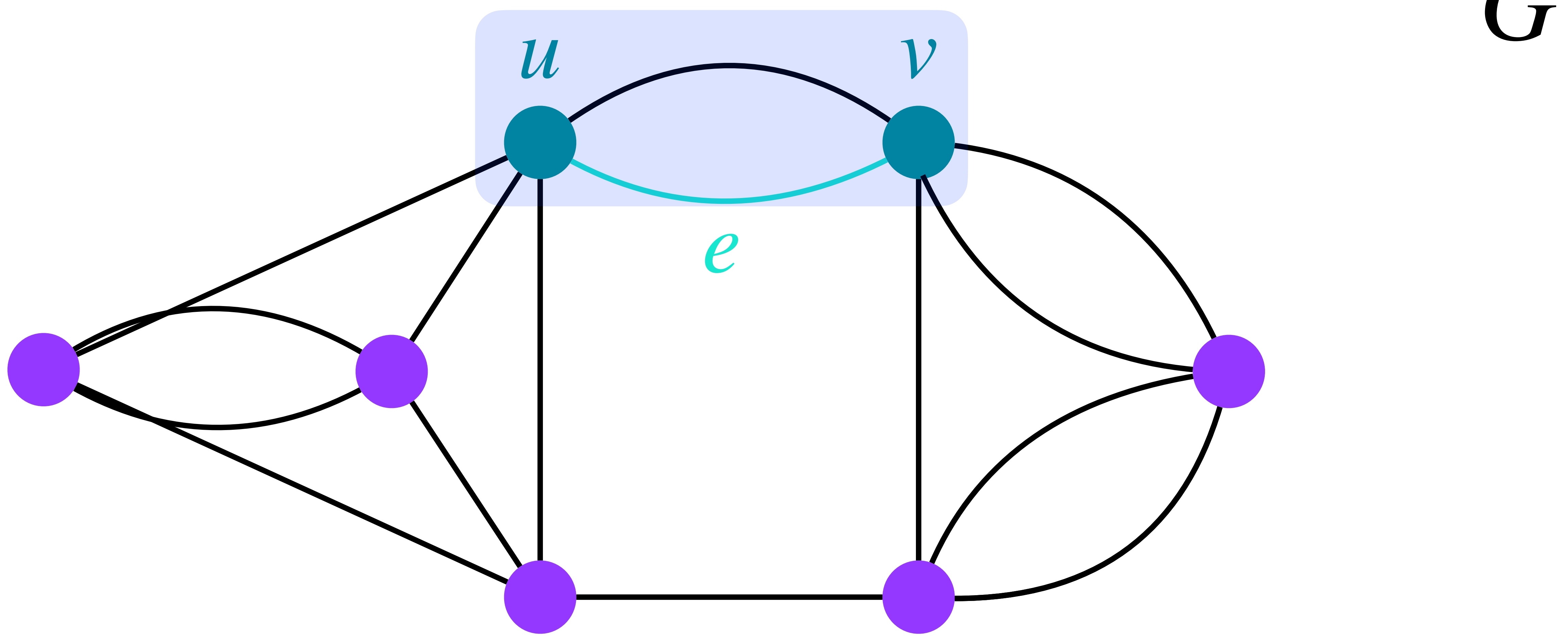
$e = \{u, v\} \in E$



Min-Cut

Edge Contraction of e

$e = \{u, v\} \in E$



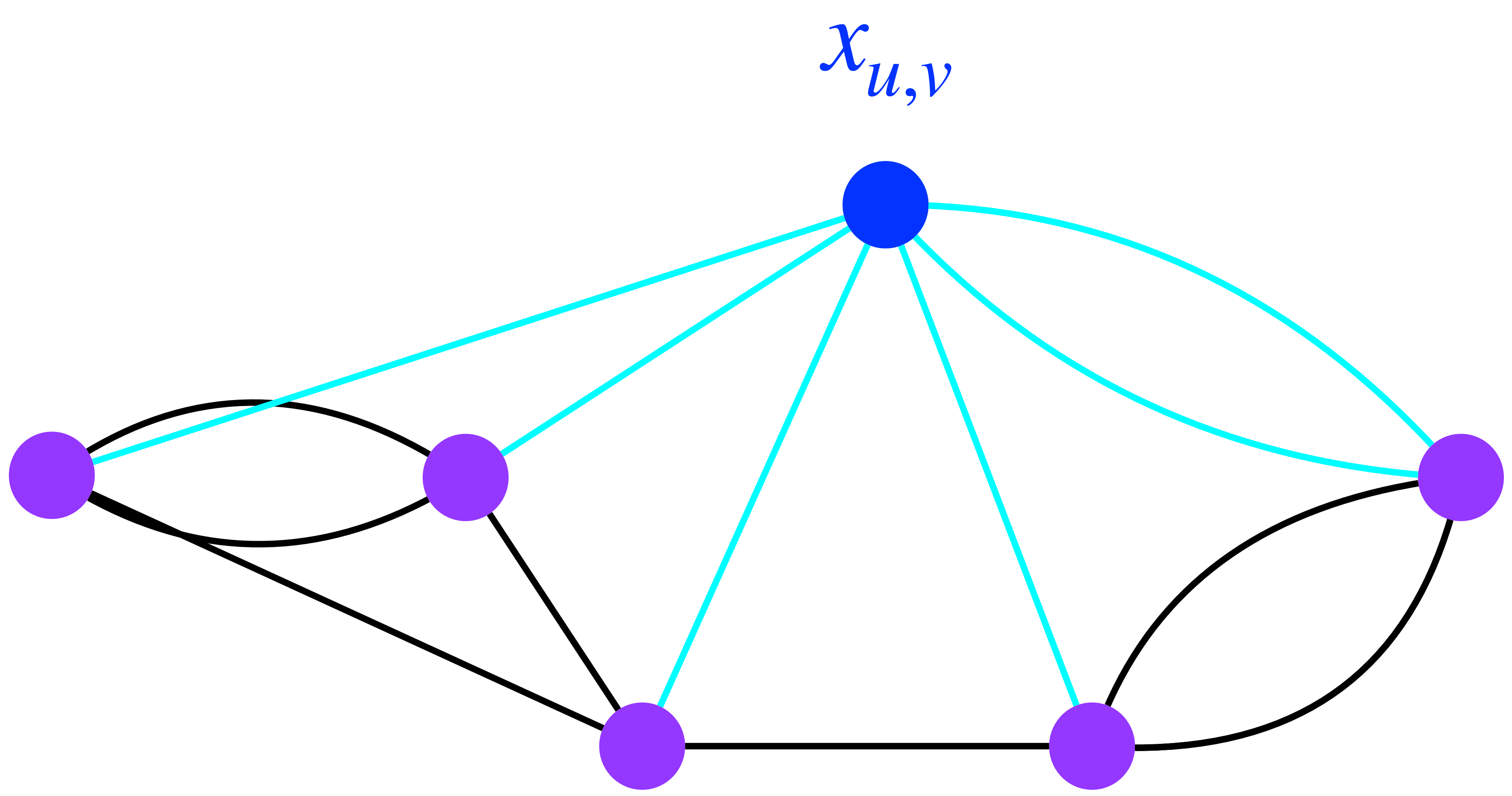
For $w, w' \in V(G) \setminus \{u, v\}$:

$$\{w, w'\} \mapsto \{w, w'\}, \quad \{w, u\} \mapsto \{w, x_{u,v}\}, \quad \{w, v\} \mapsto \{w, x_{u,v}\}$$

Min-Cut

Edge Contraction of e

$e = \{u, v\} \in E$



G / e

For $w, w' \in V(G) \backslash \{u, v\}$:

$$\{w, w'\} \mapsto \{w, w'\}, \quad \{w, u\} \mapsto \{w, x_{u,v}\}, \quad \{w, v\} \mapsto \{w, x_{u,v}\}$$

Min-Cut

Lemma

- $\mu(G)$:
  - the cardinality of the smallest possible edge cut in graph G.

Let $G = (V, E)$ be a multigraph, $e \in E$

$$\mu(G \backslash e) \geq \mu(G)$$

If G has a minimum cut C s.t. $e \notin C$

$$\mu(G \backslash e) = \mu(G)$$

Find e whose contraction preserves μ

The minimum cut value μ can never decrease when contracting an edge

μ stays unchanged if there exists a minimum cut that doesn't contain the edge being contracted

Min-Cut

Cut(G)

```
G' ← G
while |V(G')| > 2 do
    e ← uniformly random edge in G'
    G' ← G'\e
return size of the unique cut in G'
```

Runtime : $O(n^2)$

Min-Cut

Cut(G)

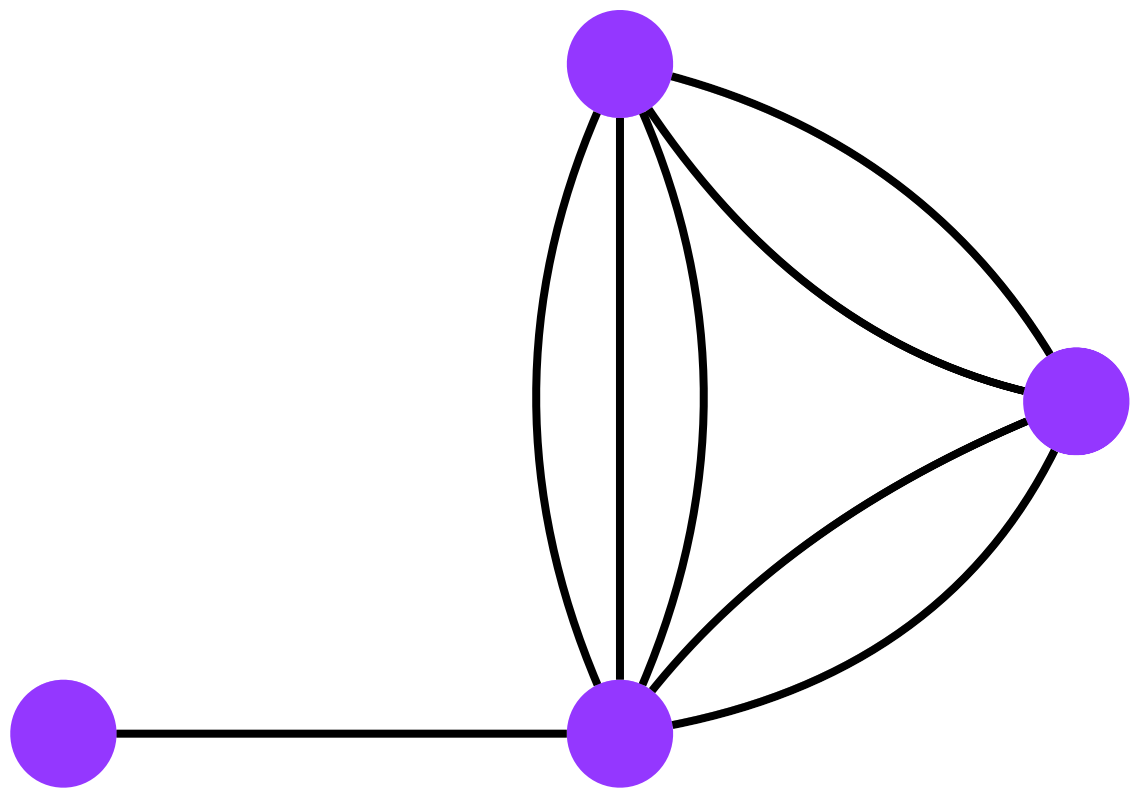
$G' \leftarrow G$

while $|V(G')| > 2$ do

    $e \leftarrow$ uniformly random edge in G'

    $G' \leftarrow G' \backslash e$

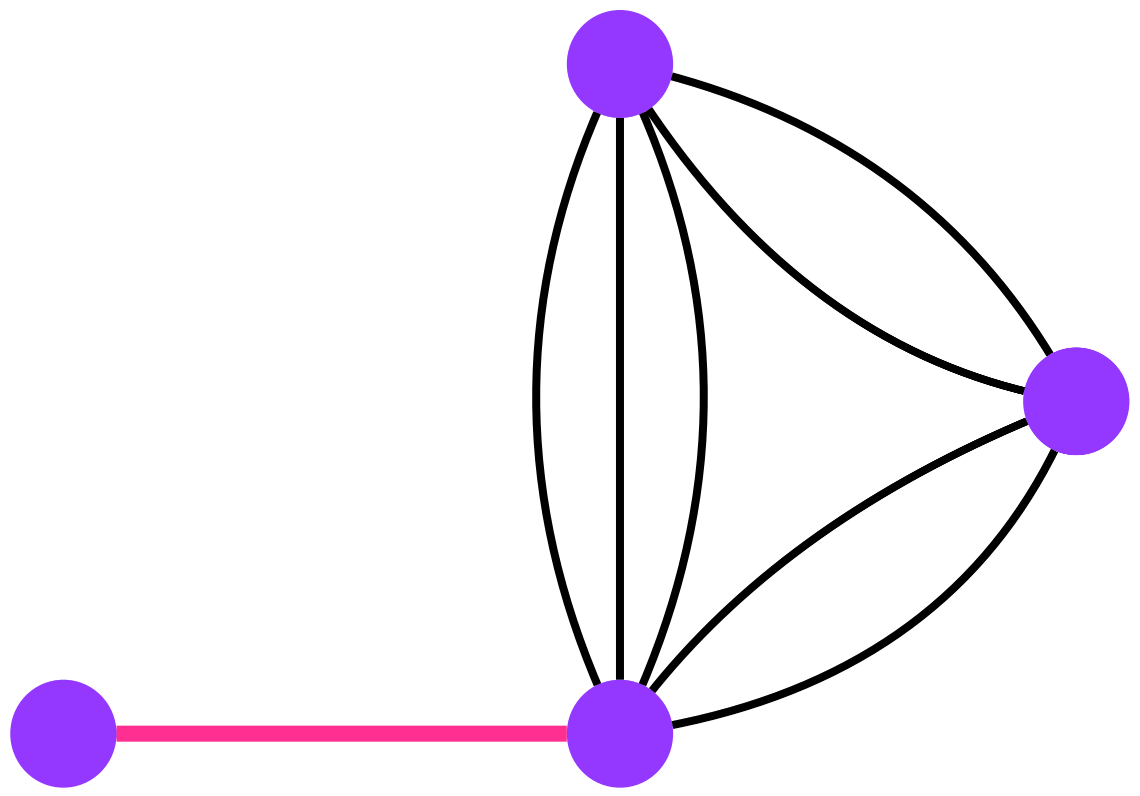
return size of the unique cut in G'



Min-Cut

Cut(G)

```
G' ← G
while |V(G')| > 2 do
    e ← uniformly random edge in G'
    G' ← G'\e
return size of the unique cut in G'
```



Min-Cut

Cut(G)

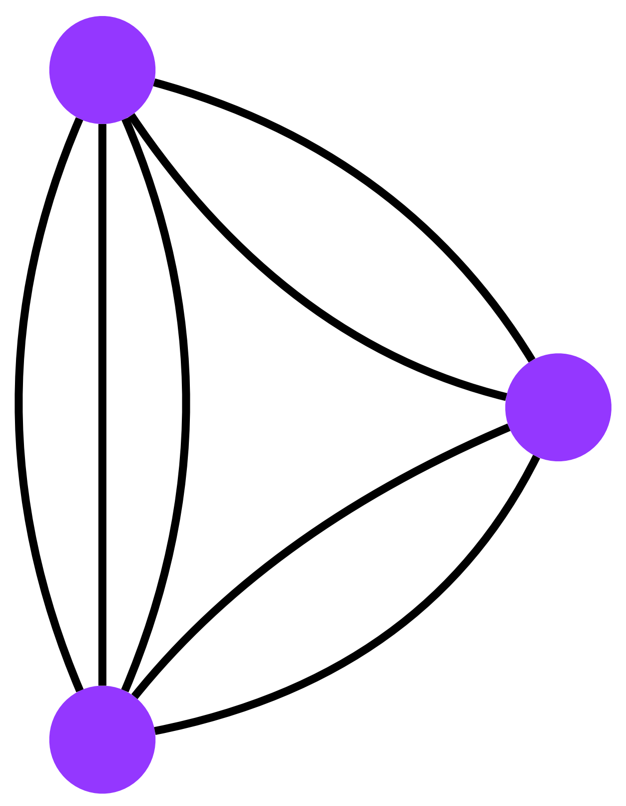
$G' \leftarrow G$

while $|V(G')| > 2$ do

    $e \leftarrow$ uniformly random edge in G'

    $G' \leftarrow G' \backslash e$

return size of the unique cut in G'



Min-Cut

Cut(G)

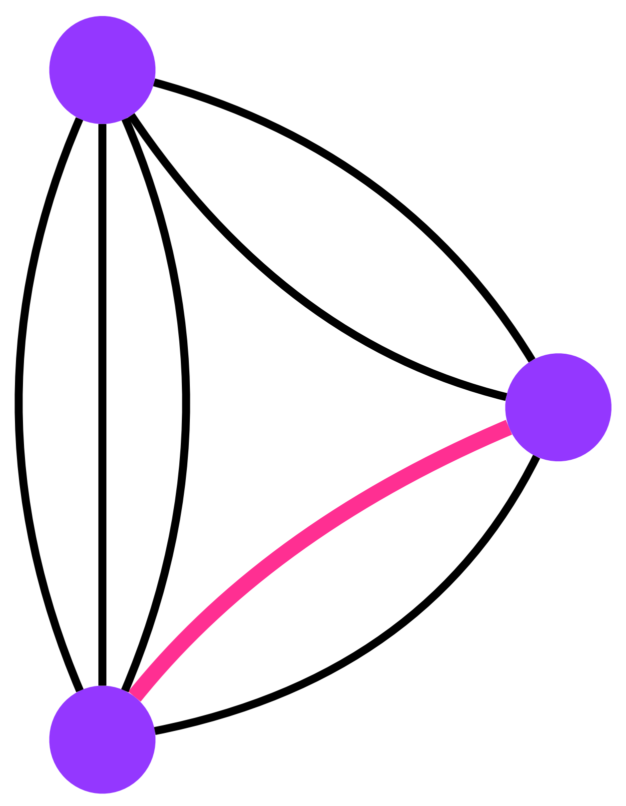
$G' \leftarrow G$

while $|V(G')| > 2$ do

    $e \leftarrow$ uniformly random edge in G'

    $G' \leftarrow G' \setminus e$

return size of the unique cut in G'



Min-Cut
Cut(G)

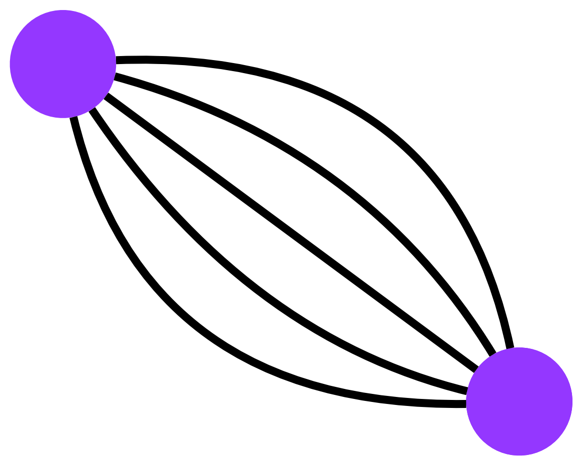
$G' \leftarrow G$

while $|V(G')| > 2$ do

    $e \leftarrow$ uniformly random edge in G'

    $G' \leftarrow G' \backslash e$

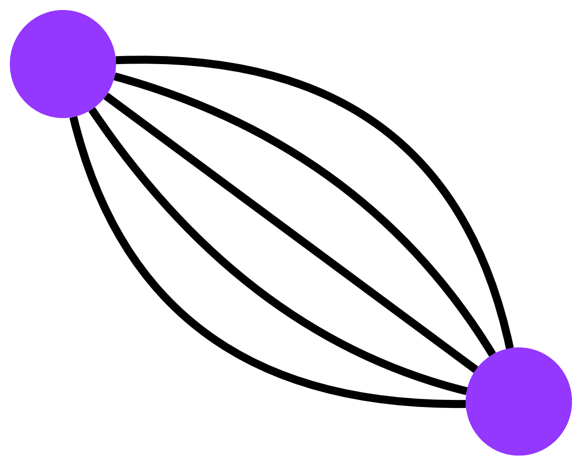
return size of the unique cut in G'



Min-Cut

Cut(G)

```
G′ ← G
while |V(G′)| > 2 do
    e ← uniformly random edge in G′
    G′ ← G′\e
return size of the unique cut in G′
```



5 !

Min-Cut

Cut(G)

For an edge e :

$$Pr[\mu(G) = \mu(G/e)] \geq 1 - \frac{2}{n}$$

For all G with $|V| = n$, $n \geq 3$:

$\hat{p}(G)$:= Probability that Cut(G) returns the value $\mu(G)$

$$\hat{p}(n) := \inf_{\substack{G=(V,E),\\ |V|=n}} \hat{p}(G) \qquad \hat{p}(n) \geq \left(1 - \frac{2}{n}\right) \cdot \hat{p}(n-1) \qquad \hat{p}(n) \geq \frac{2}{n(n-1)} = \frac{1}{\binom{n}{2}}$$

Min-Cut

Cut(G)

We repeat the algorithm Cut(G) $\lambda \binom{n}{2}$ times for some $\lambda > 0$ and return the smallest value obtained.

Runtime : O(λn^4)

Success Probability : The smallest encountered value equals $\mu(G)$ with probability at least $1 - e^{-\lambda}$

$\lambda := lnn$, runtime is O($n^4 logn$) with failure probability $\leq 1/n$

we already had a deterministic solution with this runtime !

Min-Cut

Idea : Last steps are critical

Cut(G) + Strategy Switch in the Critical Region

Stop contracting when there are t vertices remaining

switch to a randomized O(t^4) algorithm with success probability $\geq 1 - e^{-1}$

Runtime : $O(\lambda(\frac{n^4}{t^2} + n^2 t^2) \overset{t=\sqrt{n}}{=} O(\lambda n^3)$

Success Probability : $\geq 1 - e^{-1}$

Bootstrapping : We can use the same method to improve further. In "Limit" we have a O(n^2polylog(n)) algorithm.

Let's take a break

Smallest Enclosing Circle

Smallest Enclosing Circle
Problem Description

given : A finite set of points $P \subseteq \mathbb{R}^2$

to find : The circle with the smallest radius that encloses all points in P

C encloses P :

$C' :=$ the closed disk bounded by C

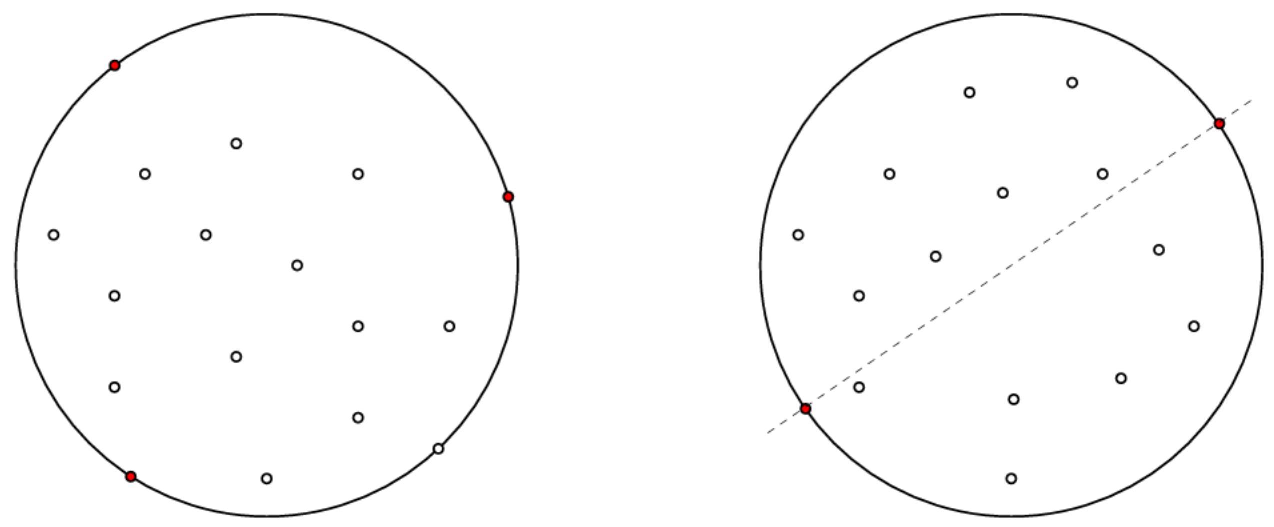
C encloses P if $P \subseteq C'$

Smallest Enclosing Circle

Lemmas

For every finite set of points $P \subseteq \mathbb{R}^2$ there exists a unique smallest enclosing cycle $C(P)$

For every finite set of points $P \subseteq \mathbb{R}^2$ with $|P| \geq 3$ there exists a subset $Q \subseteq P$ with $|Q| = 3$ s.t. $C(Q) = C(P)$



Q acts as a certificate for $C(P)$

Smallest Enclosing Circle

Easy Algorithm

Erster einfacher Algorithmus

$\left|\binom{P}{3}\right| = O(n^3)$ $(|P| = n)$

CompleteEnumeration(P)

```
for all Q ∈ (P choose 3) do
    bestimme C(Q)
    if P ⊆ C•(Q) then
        return C(Q)
```

Für alle Teilmengen $Q \subseteq P$ mit $|Q| = 3$

Falls der kleinste umschliessende Kreis von Q alle Punkte von P enthält, dann ....

Innerer Teil vom Loop: $O(n)$

Gesamte Laufzeit $O(n^4)$

Smallest Enclosing Circle

Algorithm

Runtime : O(n log n)

$P' \leftarrow P$

repeat

    randomly and uniformly choose a subset $Q \subseteq P'$ with $|Q| = 11$

    compute $C(Q)$

    if $P \subseteq C'(Q)$ then return $C(Q)$

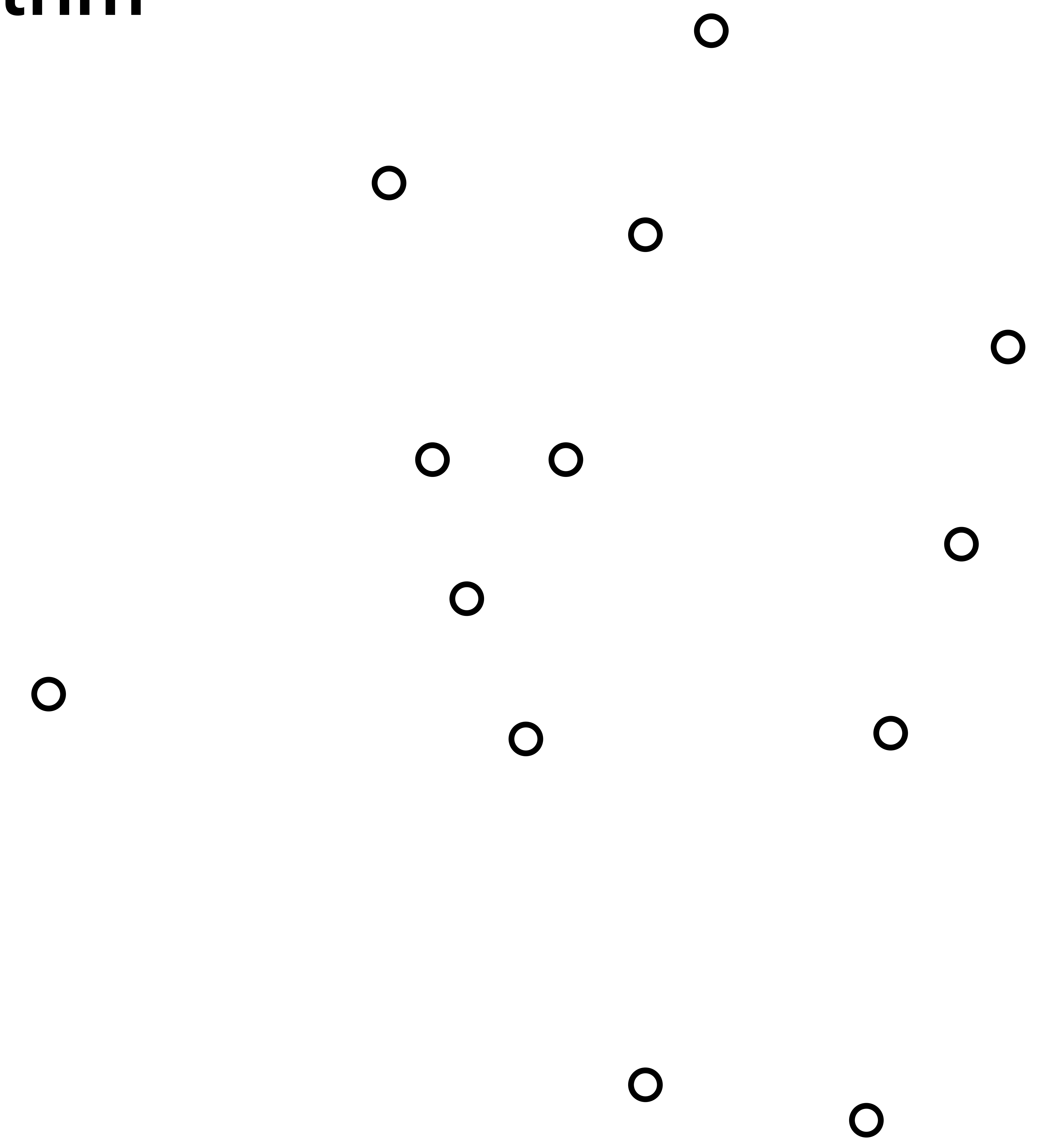
    else double all points in P' that lie outside of $C(Q)$

forever

Smallest Enclosing Circle

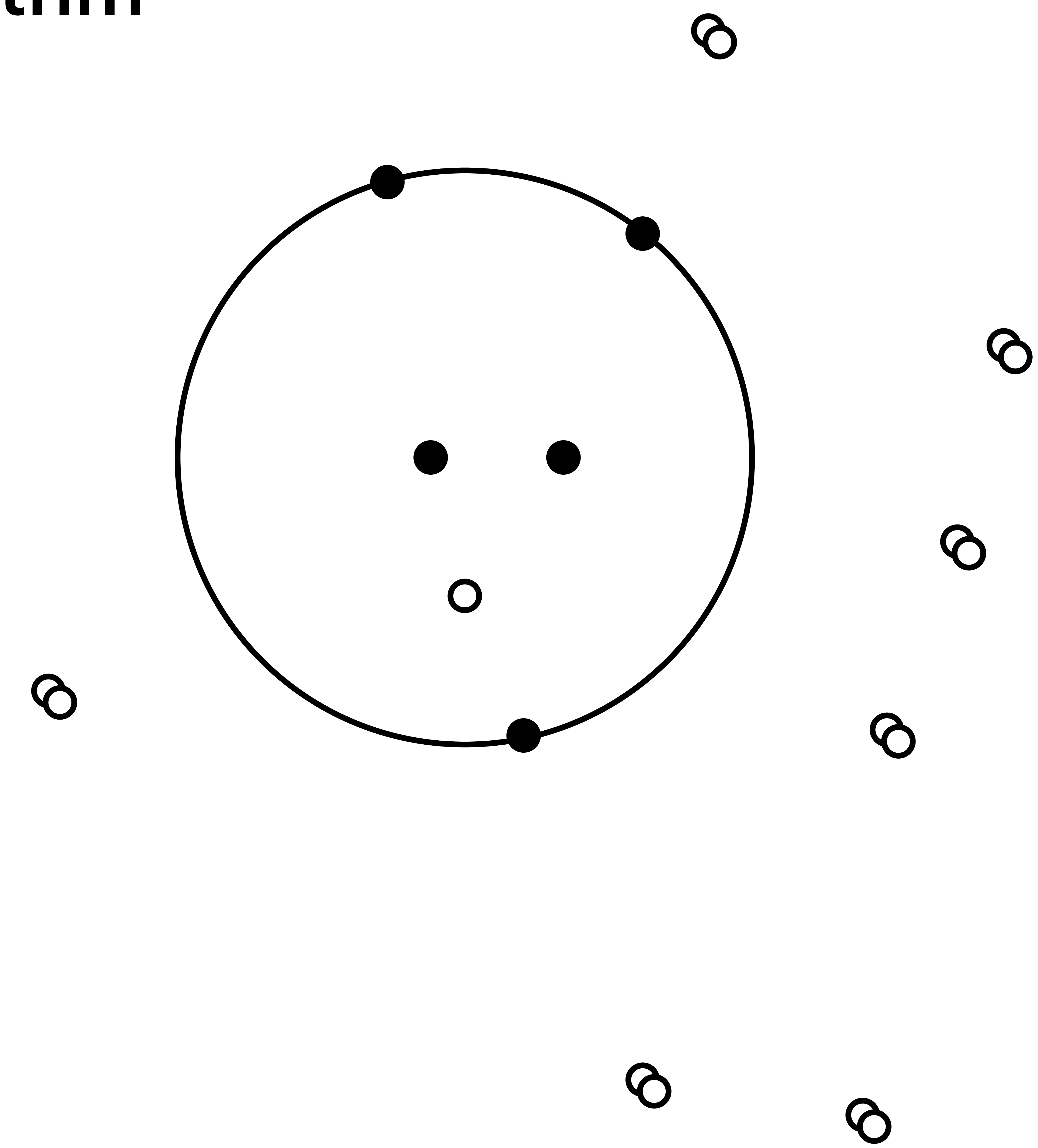
Algorithm

$P' \leftarrow P$
repeat
    randomly and uniformly choose a subset $Q \subseteq P'$ with $|Q| = 11$
    compute $C(Q)$
    if $P \subseteq C'(Q)$ then return $C(Q)$
    else double all points in P' that lie outside of $C(Q)$
forever



Smallest Enclosing Circle

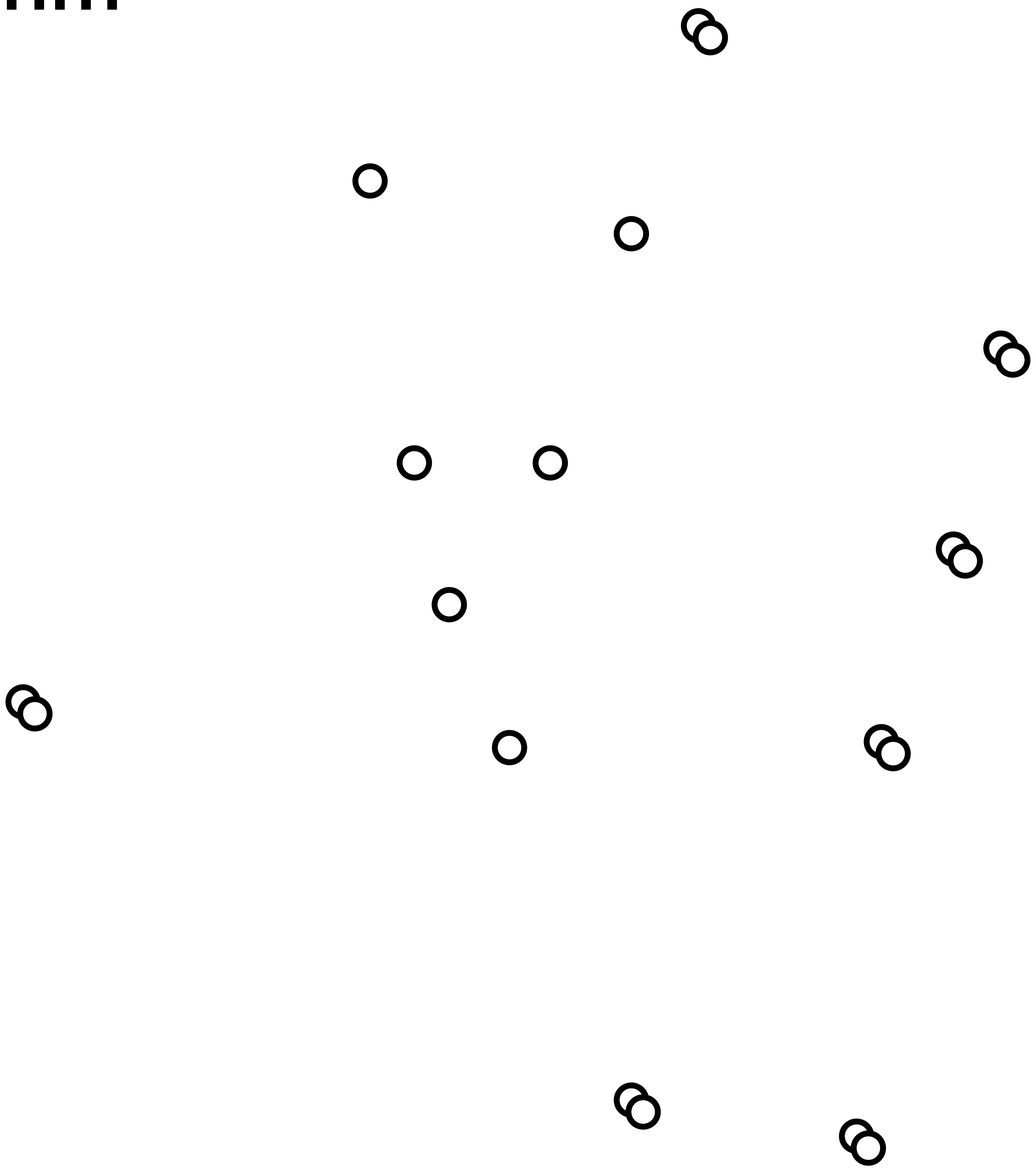
Algorithm



$P' \leftarrow P$
repeat
    randomly and uniformly choose a subset $Q \subseteq P'$ with $|Q| = 11$
    compute $C(Q)$
    if $P \subseteq C'(Q)$ then return $C(Q)$
    else double all points in P' that lie outside of $C(Q)$
forever

Smallest Enclosing Circle

Algorithm

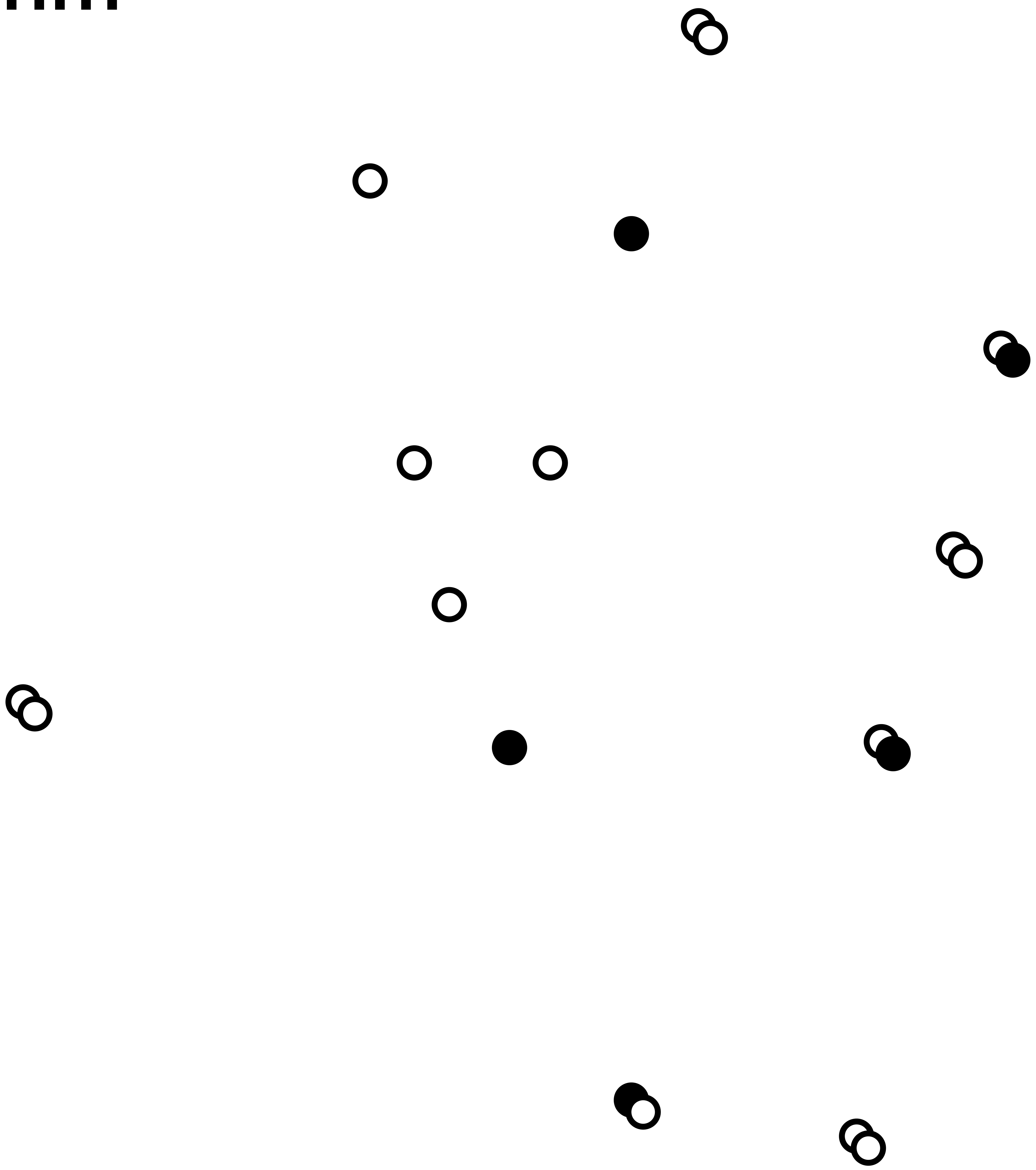


$P' \leftarrow P$
repeat
    randomly and uniformly choose a subset $Q \subseteq P'$ with $|Q| = 11$
    compute $C(Q)$
    if $P \subseteq C'(Q)$ then return $C(Q)$
    else double all points in P' that lie outside of $C(Q)$
forever

Smallest Enclosing Circle

Algorithm

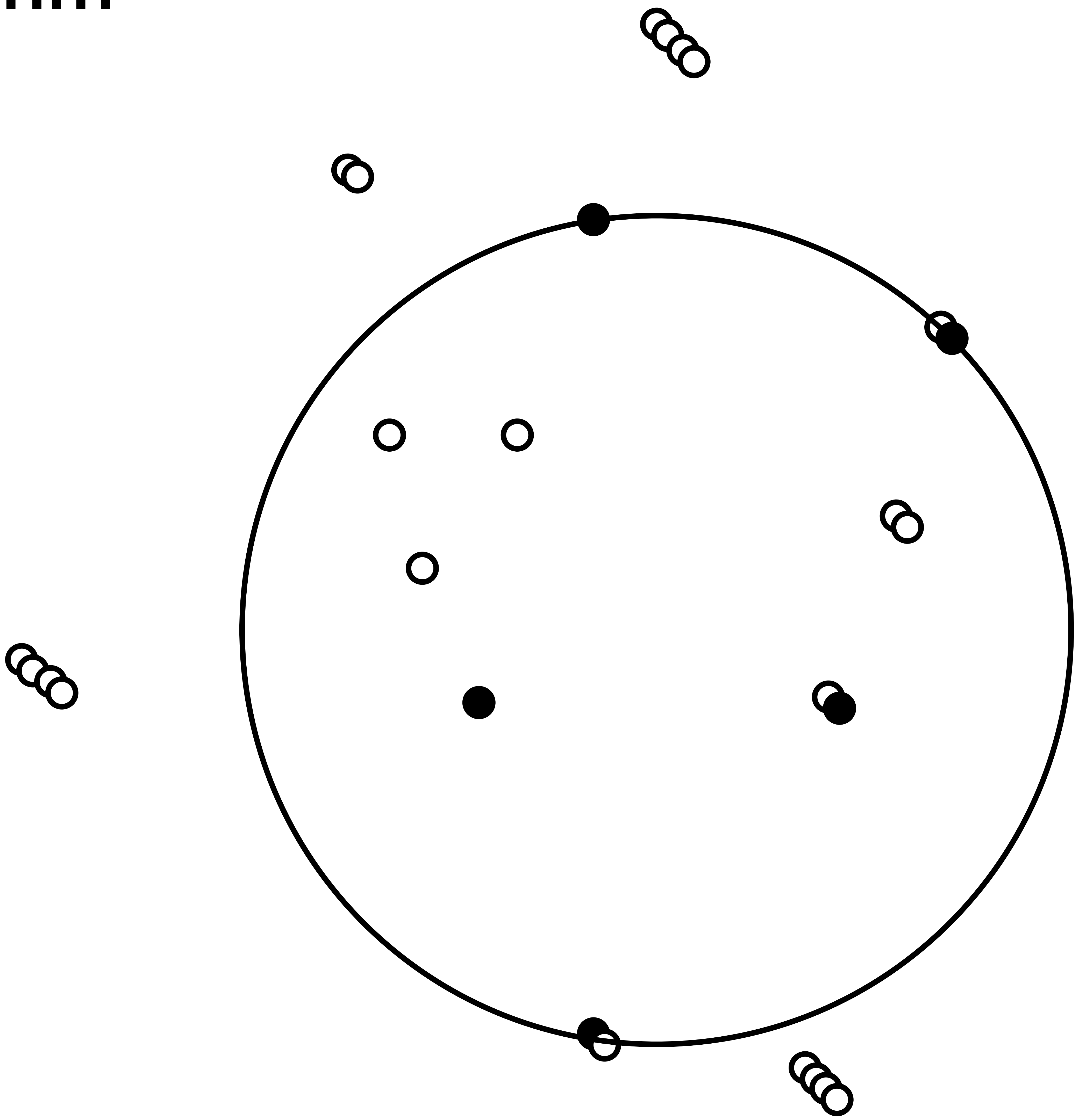
$P' \leftarrow P$
repeat
    randomly and uniformly choose a subset $Q \subseteq P'$ with $|Q| = 11$
    compute $C(Q)$
    if $P \subseteq C'(Q)$ then return $C(Q)$
    else double all points in P' that lie outside of $C(Q)$
forever



Smallest Enclosing Circle

Algorithm

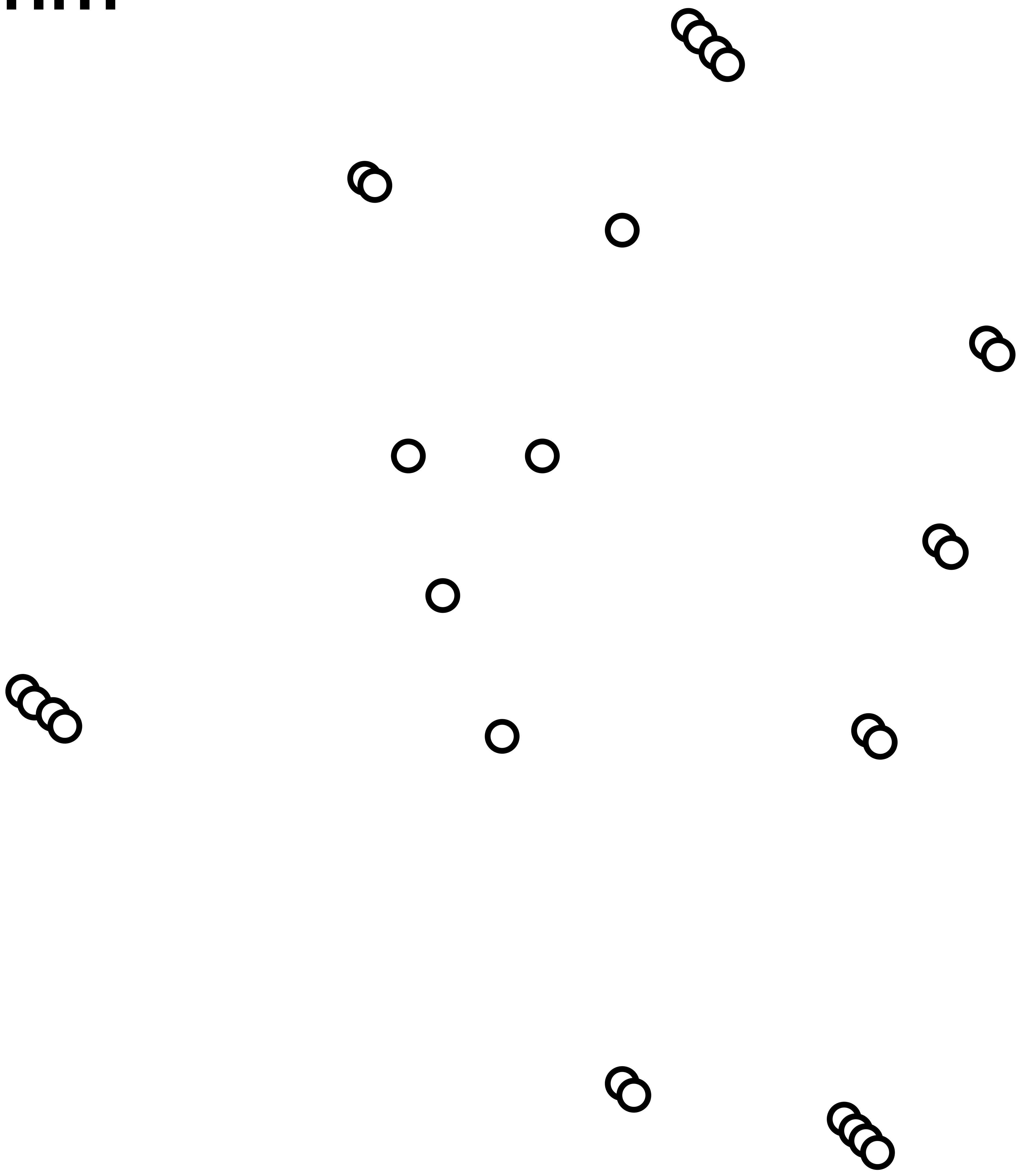
$P' \leftarrow P$
repeat
    randomly and uniformly choose a subset $Q \subseteq P'$ with $|Q| = 11$
    compute $C(Q)$
    if $P \subseteq C'(Q)$ then return $C(Q)$
    else double all points in P' that lie outside of $C(Q)$
forever



Smallest Enclosing Circle

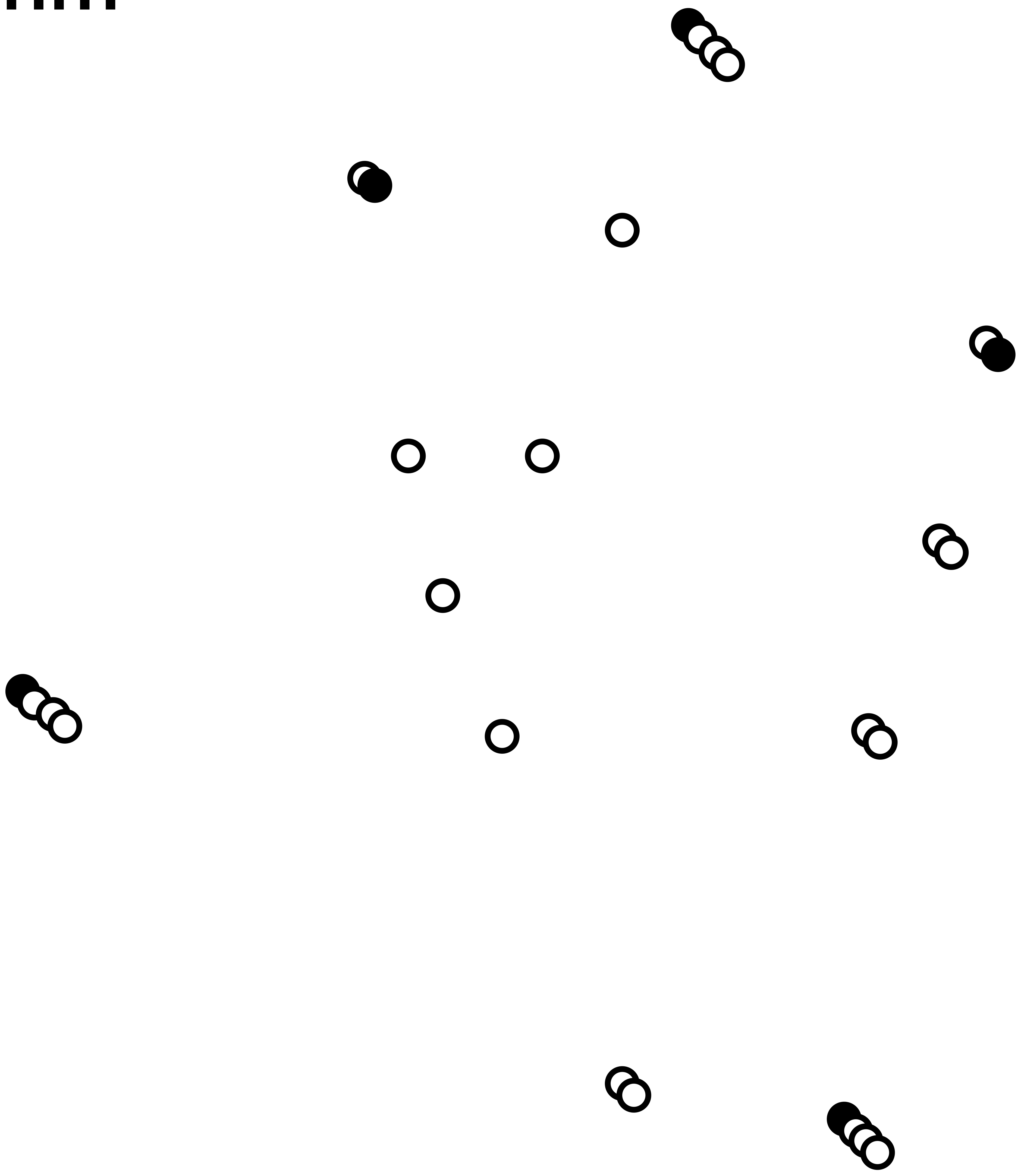
Algorithm

$P' \leftarrow P$
repeat
    randomly and uniformly choose a subset $Q \subseteq P'$ with $|Q| = 11$
    compute $C(Q)$
    if $P \subseteq C'(Q)$ then return $C(Q)$
    else double all points in P' that lie outside of $C(Q)$
forever



Smallest Enclosing Circle

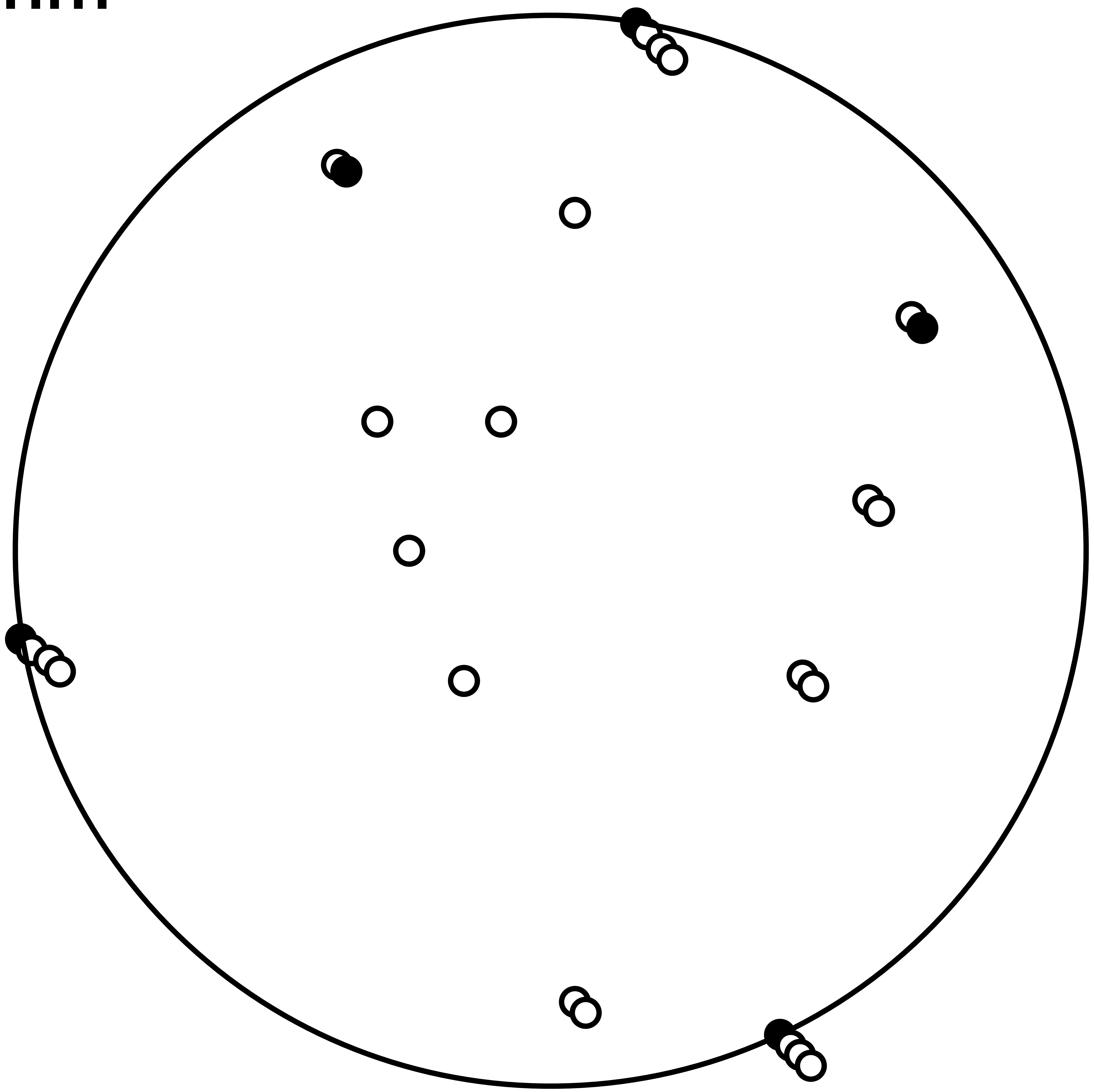
Algorithm



$P' \leftarrow P$
repeat
    randomly and uniformly choose a subset $Q \subseteq P'$ with $|Q| = 11$
    compute $C(Q)$
    if $P \subseteq C'(Q)$ then return $C(Q)$
    else double all points in P' that lie outside of $C(Q)$
forever

Smallest Enclosing Circle

Algorithm



1: $P' \leftarrow P$

2: **repeat**

3: randomly and uniformly choose a subset $Q \subseteq P'$ with $|Q| = 11$

4: compute $C(Q)$

5: if $P \subseteq C'(Q)$ then return $C(Q)$

6: else double all points in P' that lie outside of $C(Q)$

7: **forever**

Smallest Enclosing Circle

Sampling Lemma

Let $r, N \in \mathbb{N}$, $r \leq N$ and $P' \subseteq \mathbb{R}^2$ be a multiset with $|P'| = N$

For R chosen uniformly at random from $\binom{P'}{r}$, the folowing holds :

$$\mathbb{E}\left(\left|\underbrace{P' \backslash C'(R)}_{\text{Points in } P' \text{ outside } C(R)}\right|\right) \leq 3 \cdot \frac{N-r}{r+1} \leq 3 \cdot \frac{N}{r+1}$$

Minitest 5 solutions

Questions

Feedbacks , Recommendations

Nil Ozer