The Searchlight Problem for Road Networks

Dariusz Dereniowski
Hirotaka Ono
Ichiro Suzuki
Łukasz Wrona
Masafumi Yamashita
Paweł Żyliński

GRASTA-MAC 2015, Montreal

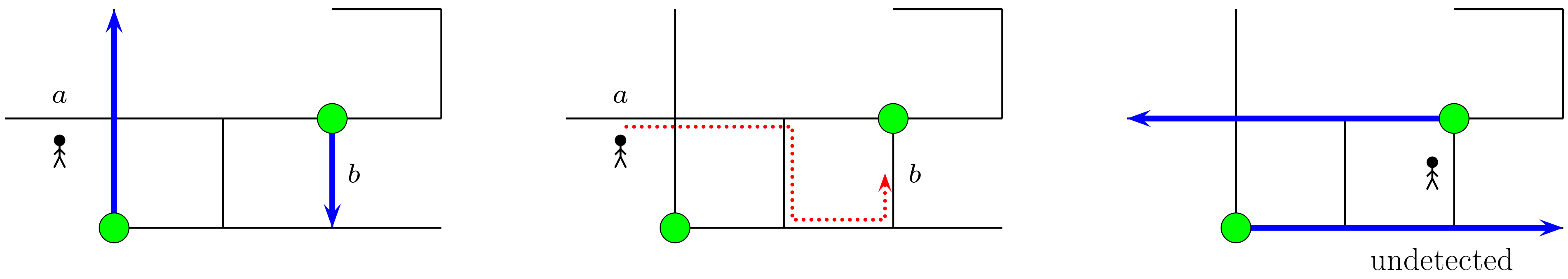
The searchlight problem in a road network

What is the worst-case number $s(n, g)$ of searchlights, each placed at one of the g guard positions, required to successfully search a given road network of n lines/line segments?

→ A mobile intruder capable of moving continuously and arbitrarily fast is hiding.

→ The objective of the guards is to detect the intruder using the rays.

→ The intruder is considered detected at the moment he is illuminated by one of the rays or he reaches a position where a guard is located.



- Sugihara, Suzuki and Yamashita. (1990): The searchlight scheduling problem
- Yen and Tang (1995): The searchlight guarding problem on weighted trees

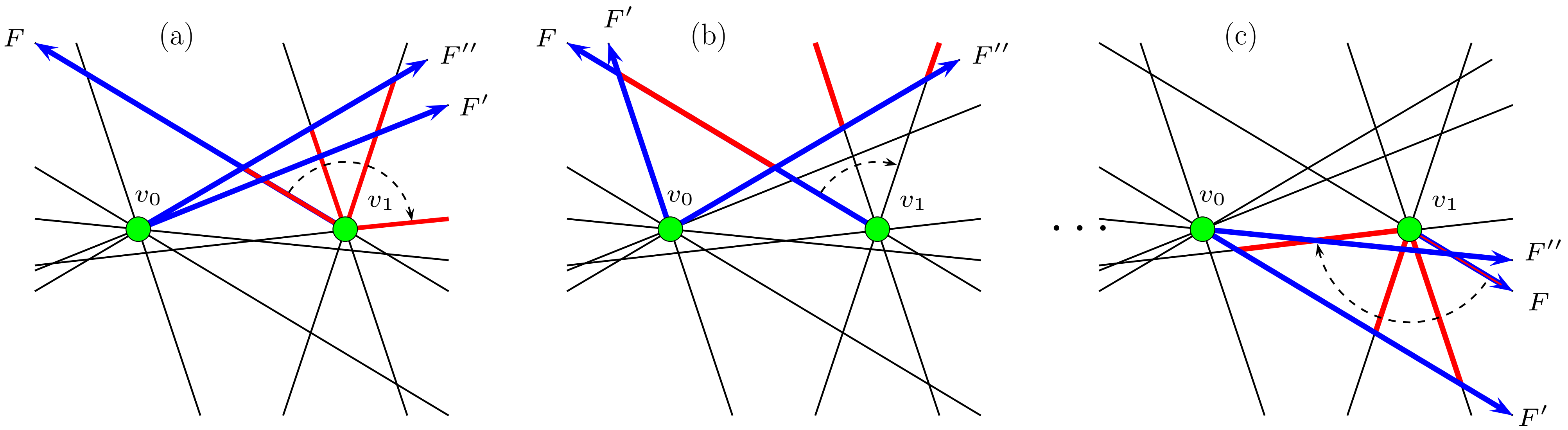
The searchlight problem in a road network

What is the worst-case number $s(n, g)$ of searchlights, each placed at one of the g guard positions, required to successfully search a given road network of n lines/line segments?

→ A mobile intruder capable of moving continuously and arbitrarily fast is hiding.

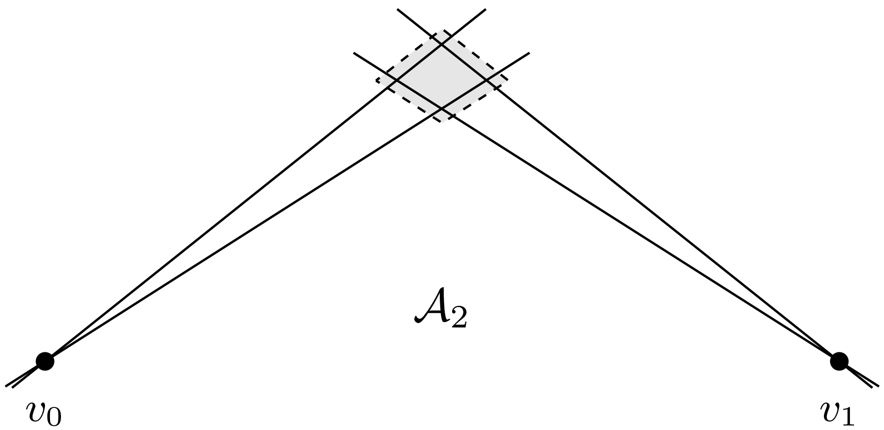
→ The objective of the guards is to detect the intruder using the rays.

→ The intruder is considered detected at the moment he is illuminated by one of the rays or he reaches a position where a guard is located.

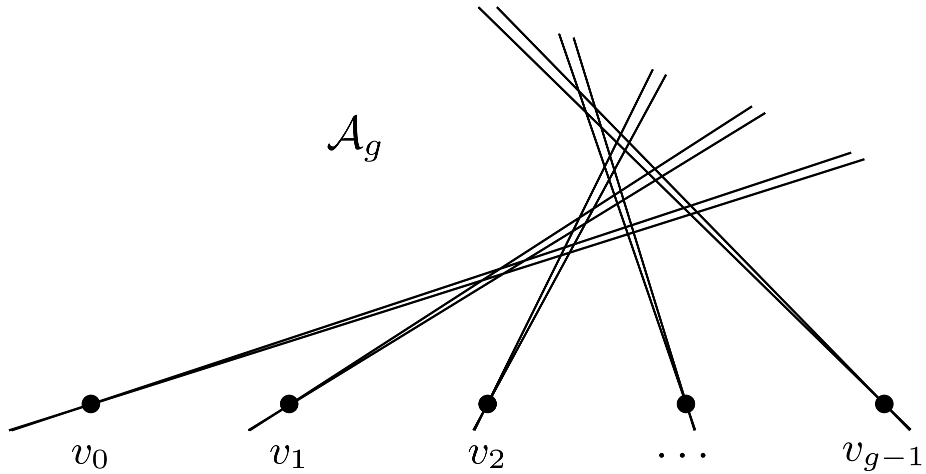


A sample search strategy for an $(n, 2)$-arrangement, $n \geq 4$.

Arrangements of lines: a lower bound of $2g-1$



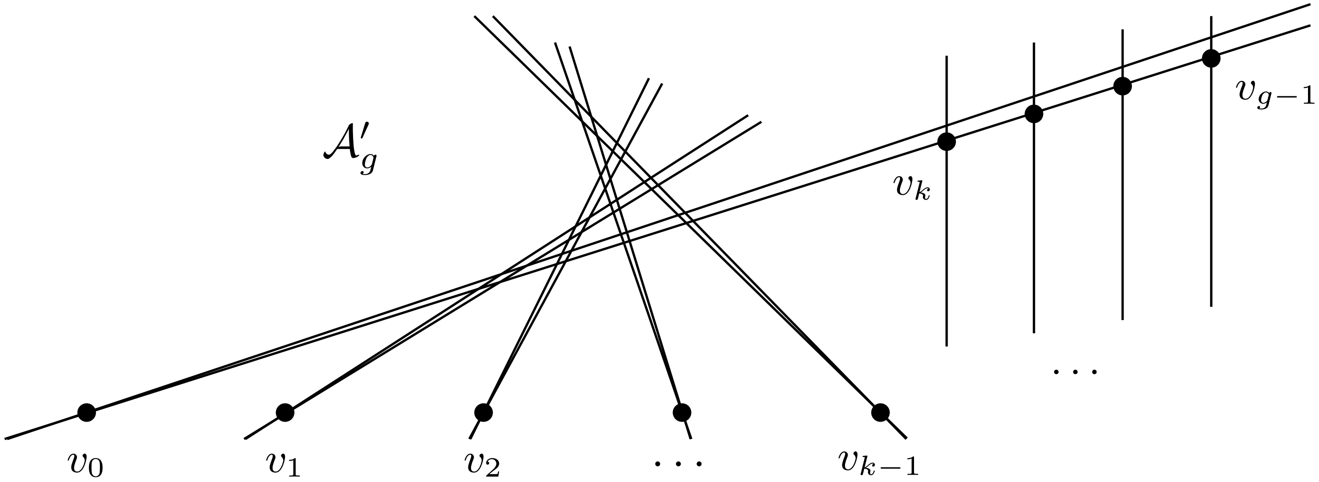
(4, 2)-arrangement $\mathcal{A}_2 = (\mathcal{L}_2, \{v_0, v_1\})$
that requires at least three searchlights



$(2g, g)$-arrangement $\mathcal{A}_g = (\mathcal{L}_g, \{v_0, \ldots, v_{g-1}\})$, $g \geq 2$

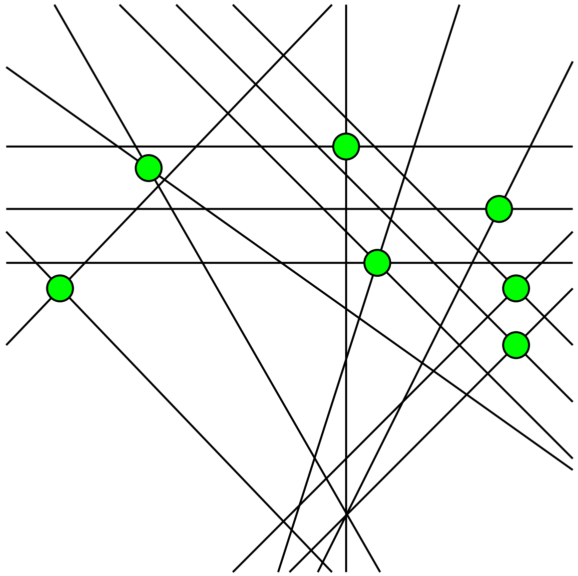
$$s(n, g) \geq 2g - 1$$

Arrangements of lines: a lower bound of $2g-1$

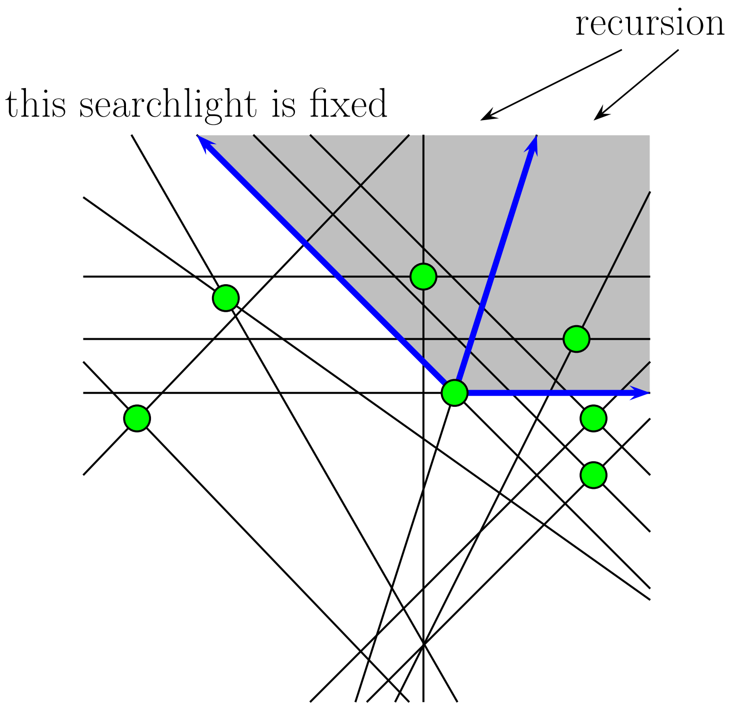


$$\text{For } 1 \le g \le n-1,\ s(n,g) \ge \begin{cases} 2g-1 \text{ if } 1 \le g \le \frac{n}{2}; \\ n-2 \quad \text{if } \frac{n}{2} < g \le n-2; \\ n-1 \quad \text{if } g = n-1. \end{cases}$$

Arrangements of lines: an upper bound of $\frac{7g}{3} - 1$

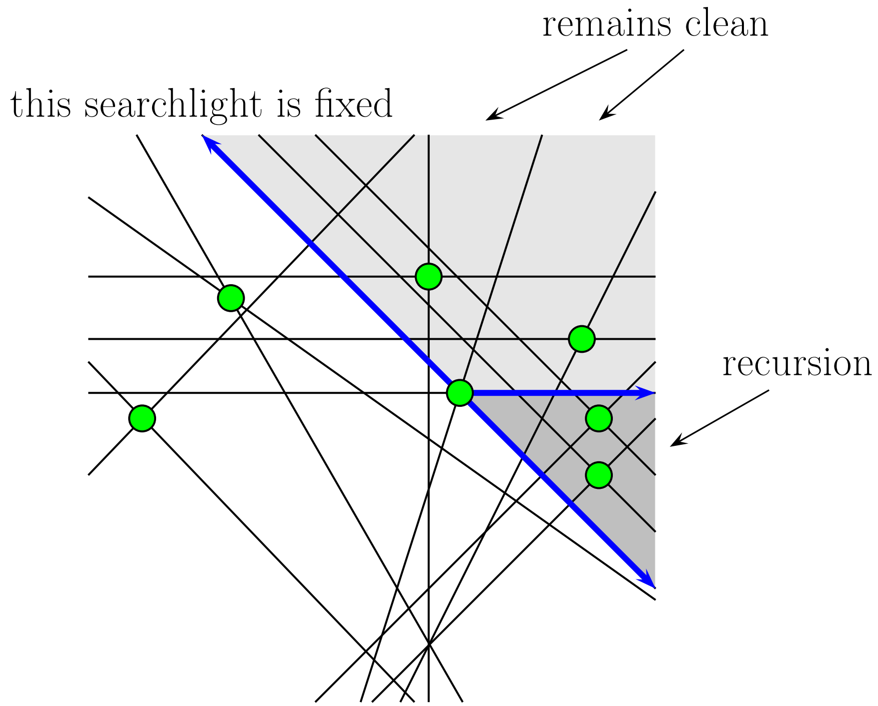


Arrangements of lines: an upper bound of $\frac{7g}{3} - 1$



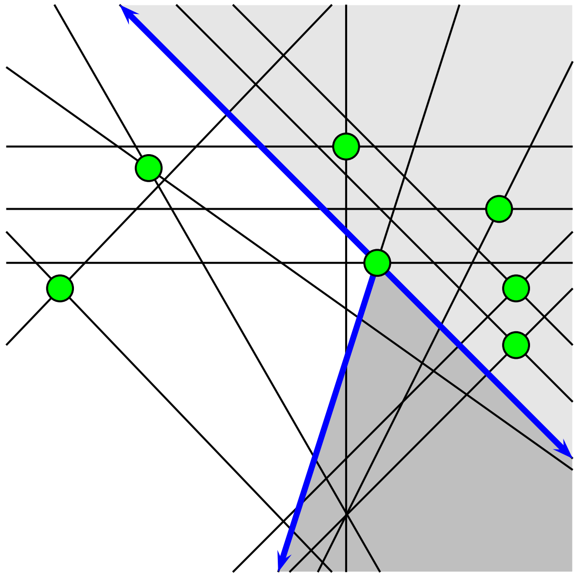
$$s(n, g) \leq 3g$$

Arrangements of lines: an upper bound of $\frac{7g}{3} - 1$



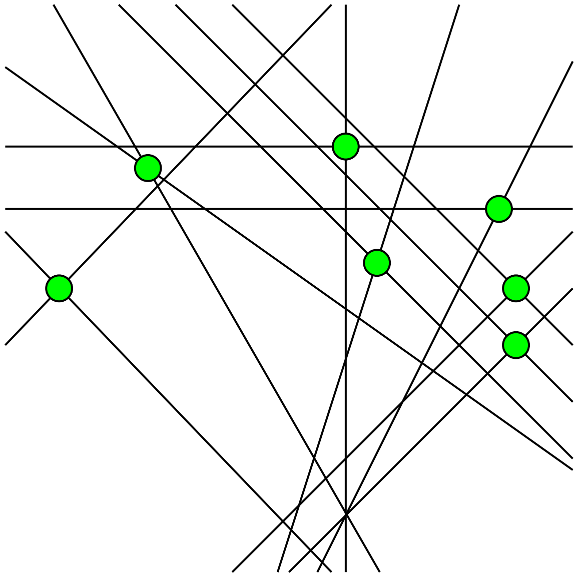
$$s(n, g) \leq 3g$$

Arrangements of lines: an upper bound of $\frac{7g}{3} - 1$

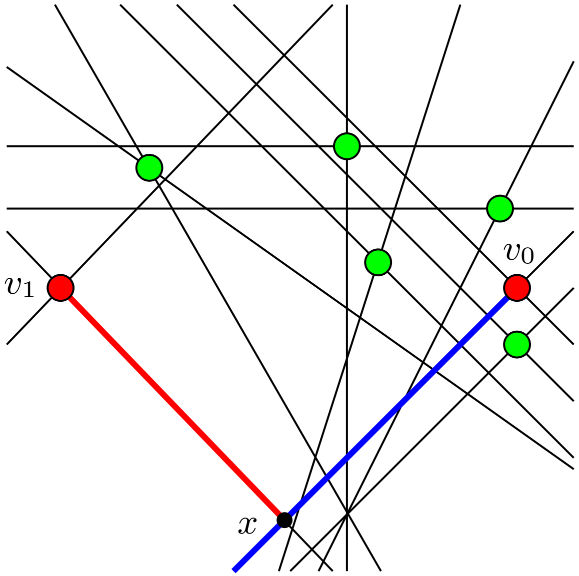


$$s(n, g) \leq 3g$$

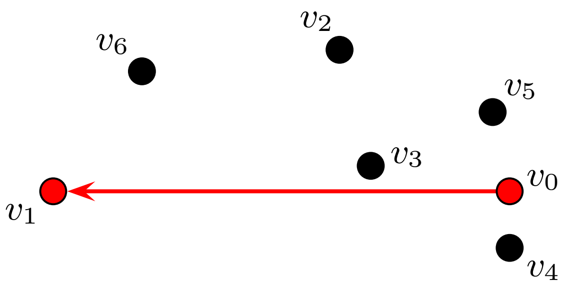
Arrangements of lines: an upper bound of $\frac{7g}{3} - 1$



Arrangements of lines: an upper bound of $\frac{7g}{3} - 1$

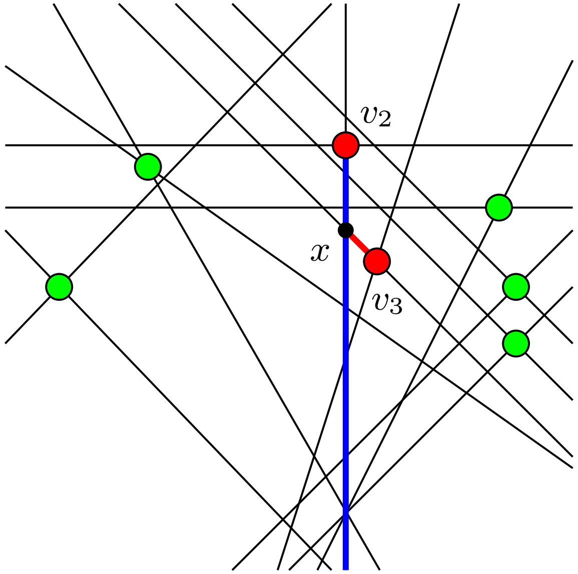


no intersection points between v_1 and x

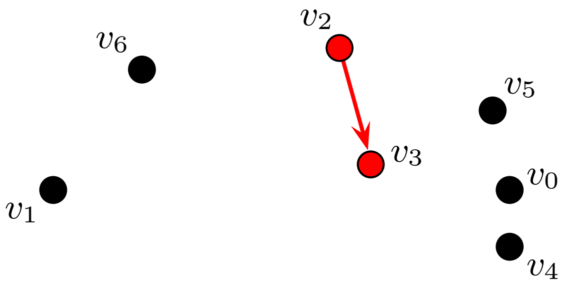


v_0 is incident to v_1

Arrangements of lines: an upper bound of $\frac{7g}{3} - 1$

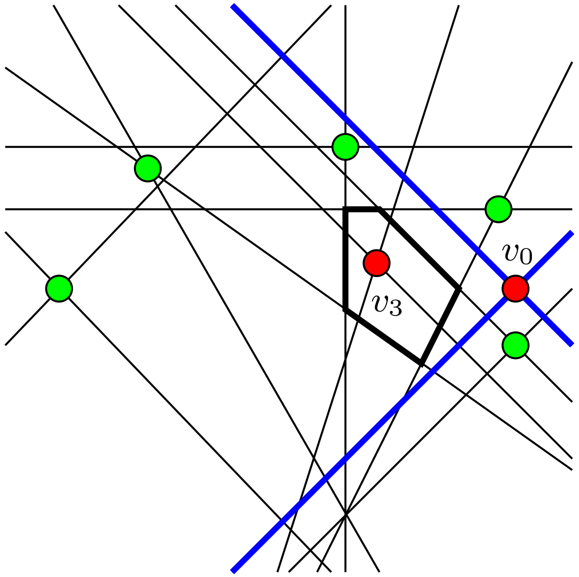


no intersection points between v_3 and x

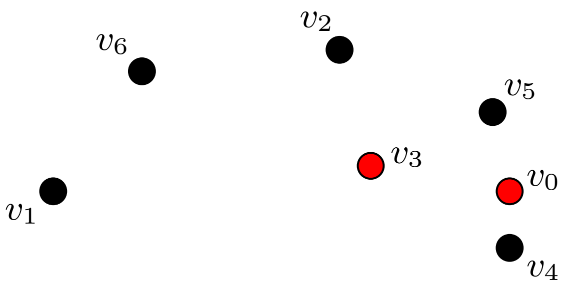


v_2 is incident to v_3

Arrangements of lines: an upper bound of $\frac{7g}{3} - 1$

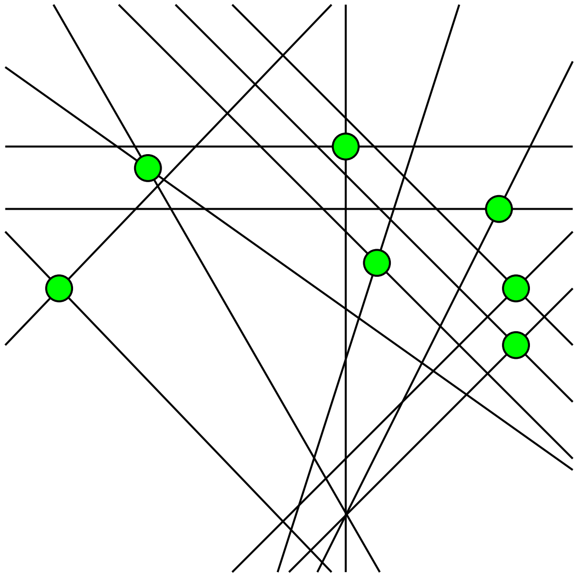


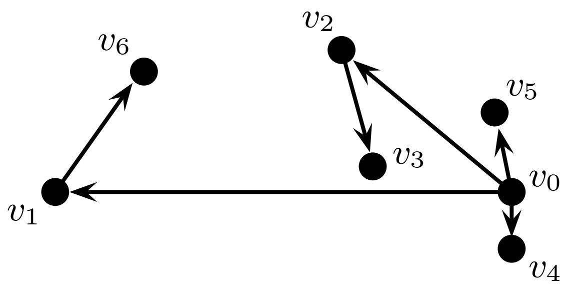
there is a 'free' cycle around v_3



v_0 is not incident to v_3

Arrangements of lines: an upper bound of $\frac{7g}{3} - 1$





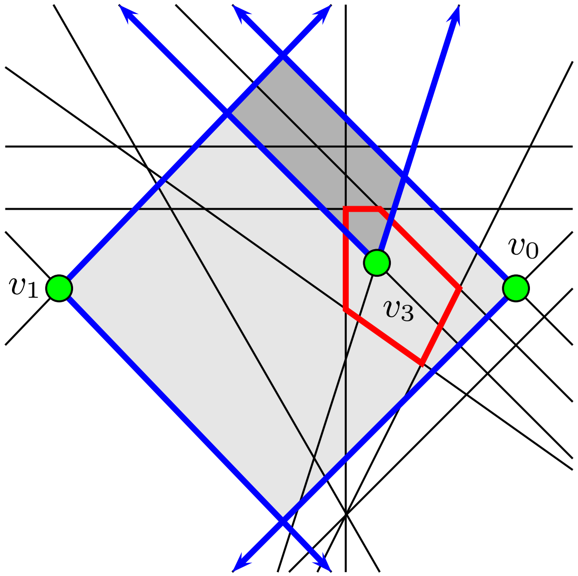
any arborescence:

3 searchlights at the root v_0

2 searchlights at any other vertex

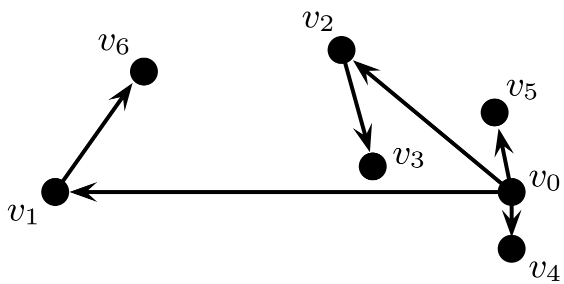
ordering: $v_0, v_1, v_2, v_3, v_4, v_5, v_6$

Arrangements of lines: an upper bound of $\frac{7g}{3} - 1$



wrong order:

v_3 is handled before handling v_2

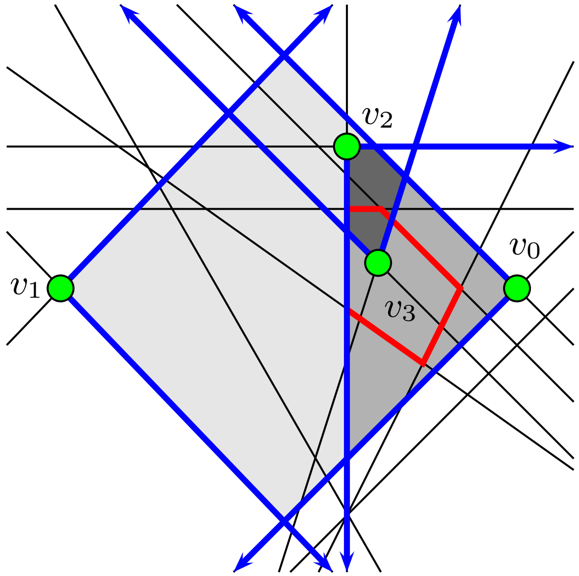


any arborescence:

3 searchlights at the root v_0

2 searchlights at any other vertex

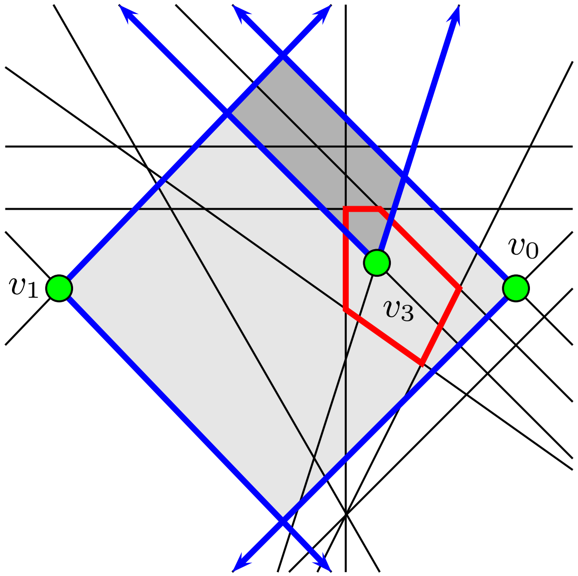
ordering: $v_0, v_1, v_2, v_3, v_4, v_5, v_6$



correct order:

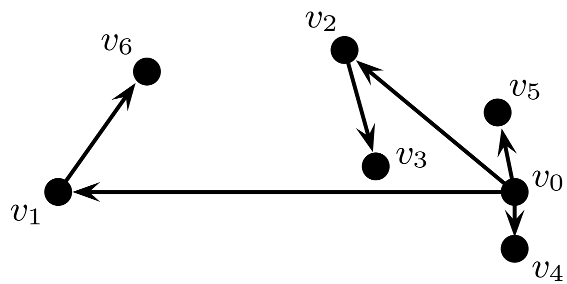
v_3 is handled after handling v_2

Arrangements of lines: an upper bound of $\frac{7g}{3} - 1$



wrong order:

v_3 is handled before handling v_2

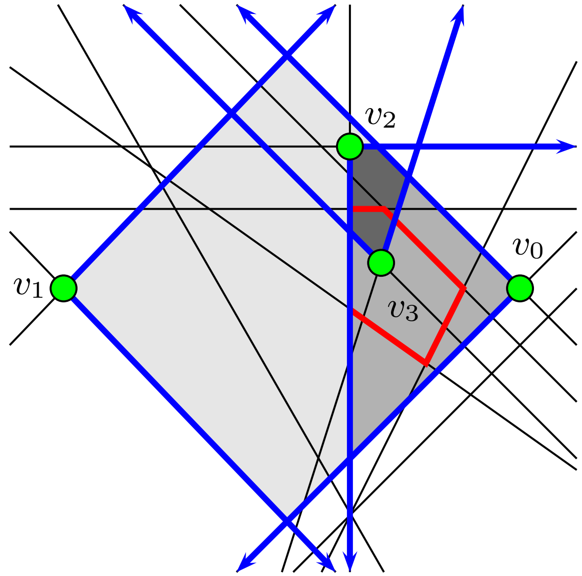


any arborescence:

3 searchlights at the root v_0

2 searchlights at any other vertex

ordering: $v_0, v_1, v_2, v_3, v_4, v_5, v_6$

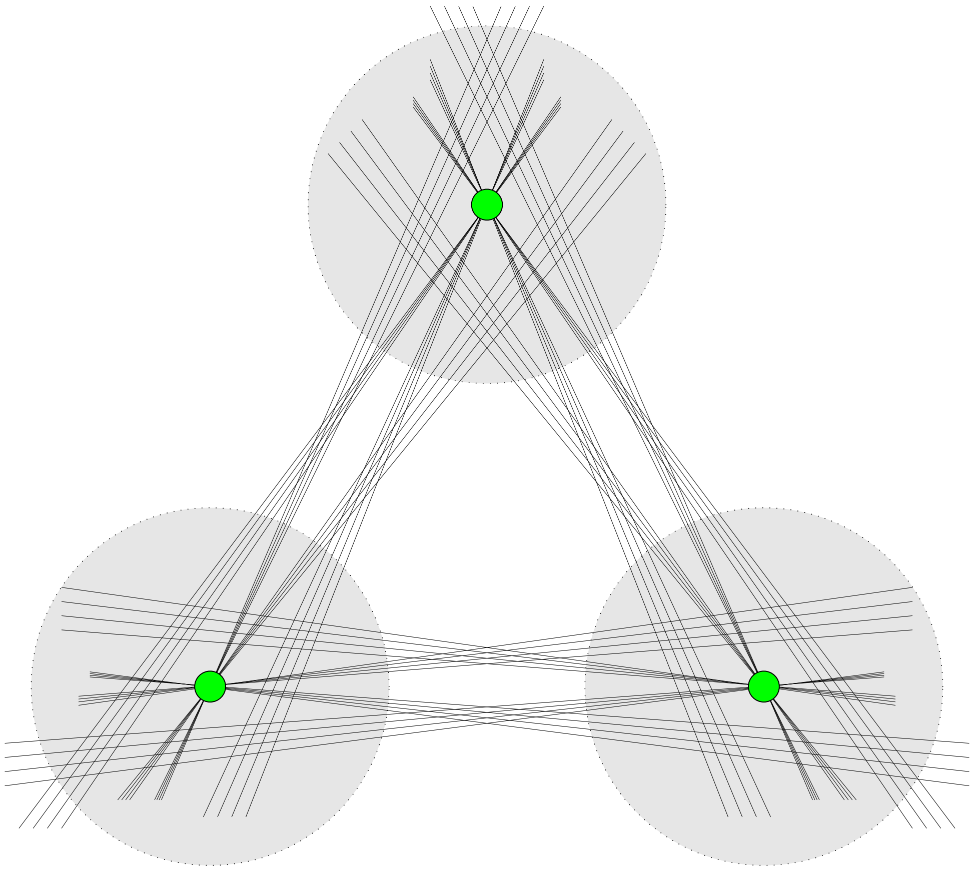


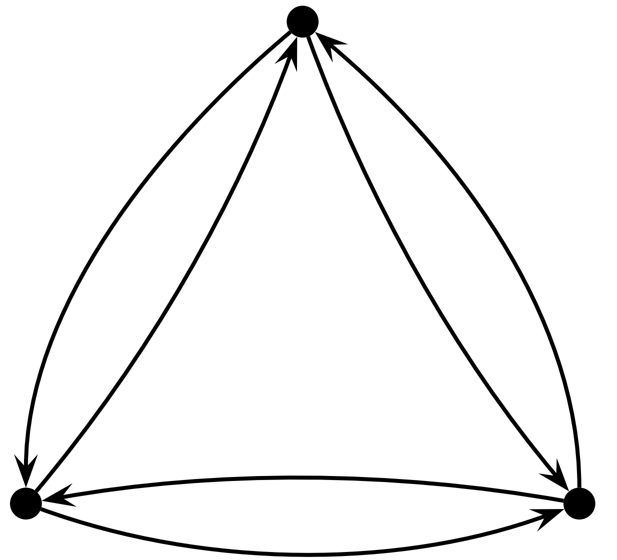
correct order:

v_3 is handled after handling v_2

$$s(\mathcal{A}) \leq 2g + (h-1).$$

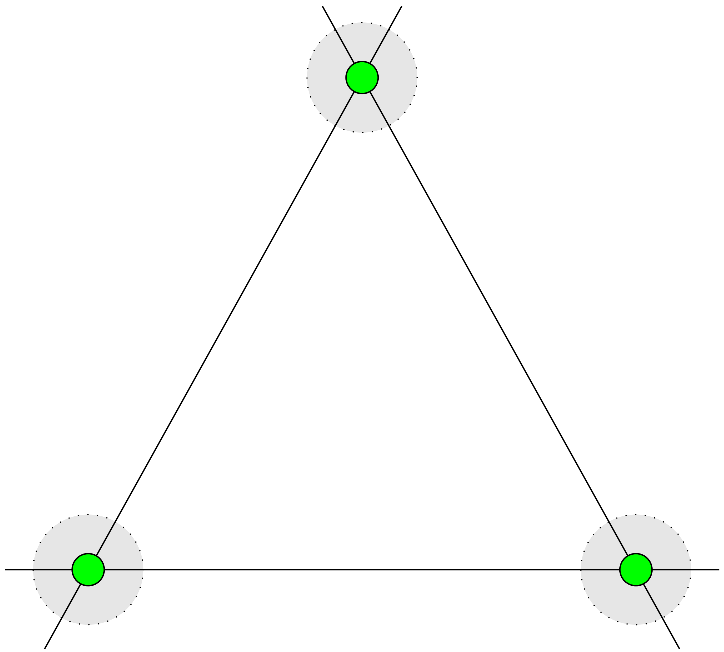
Arrangements of lines: an upper bound of $\frac{7g}{3} - 1$





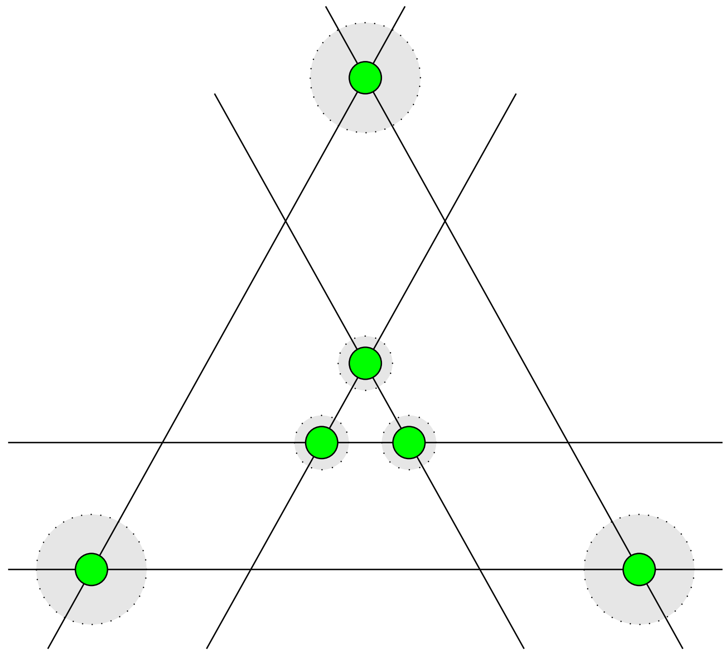
$$s(n,g) \leq \frac{7g}{3} - 1$$

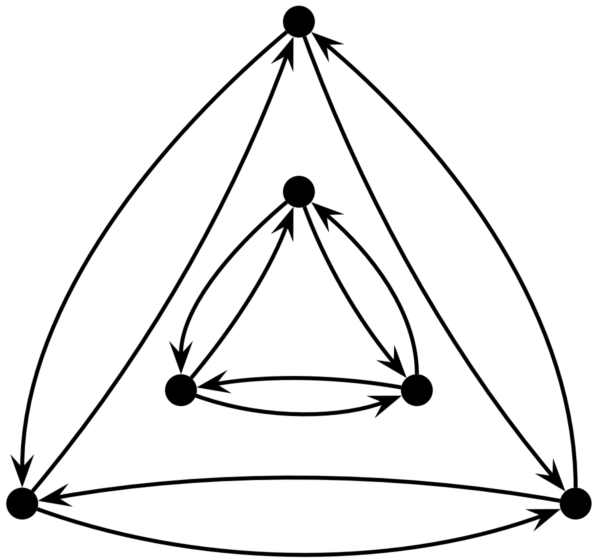
Arrangements of lines: an upper bound of $\frac{7g}{3} - 1$



$$s(n, g) \leq \frac{7g}{3} - 1$$

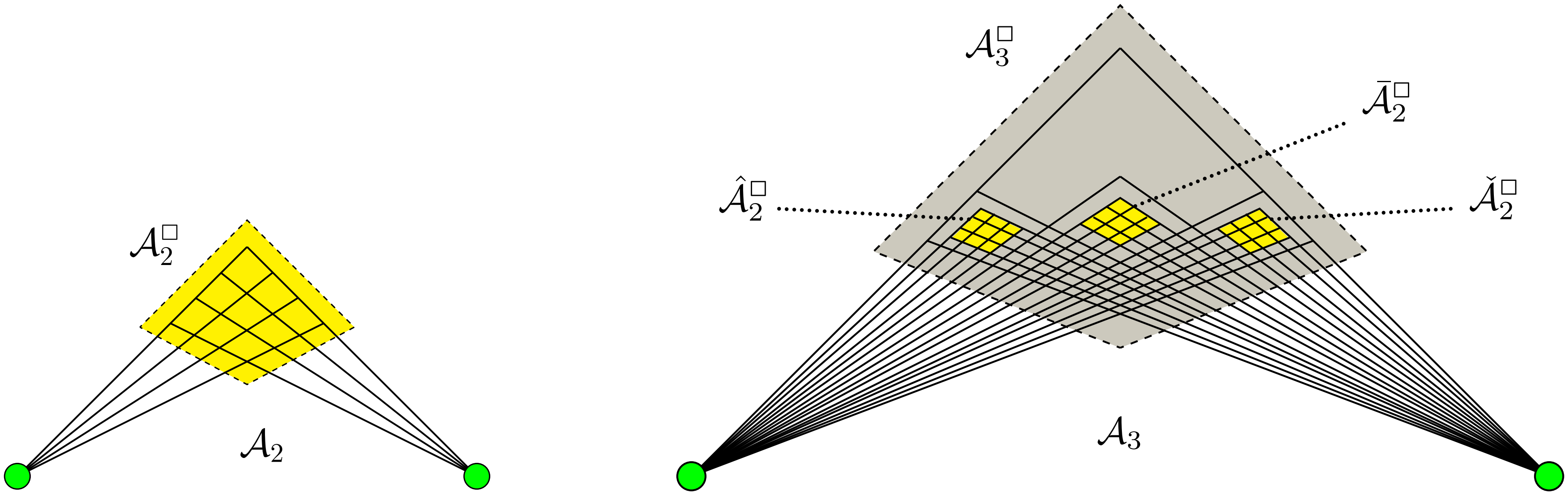
Arrangements of lines: an upper bound of $\frac{7g}{3} - 1$





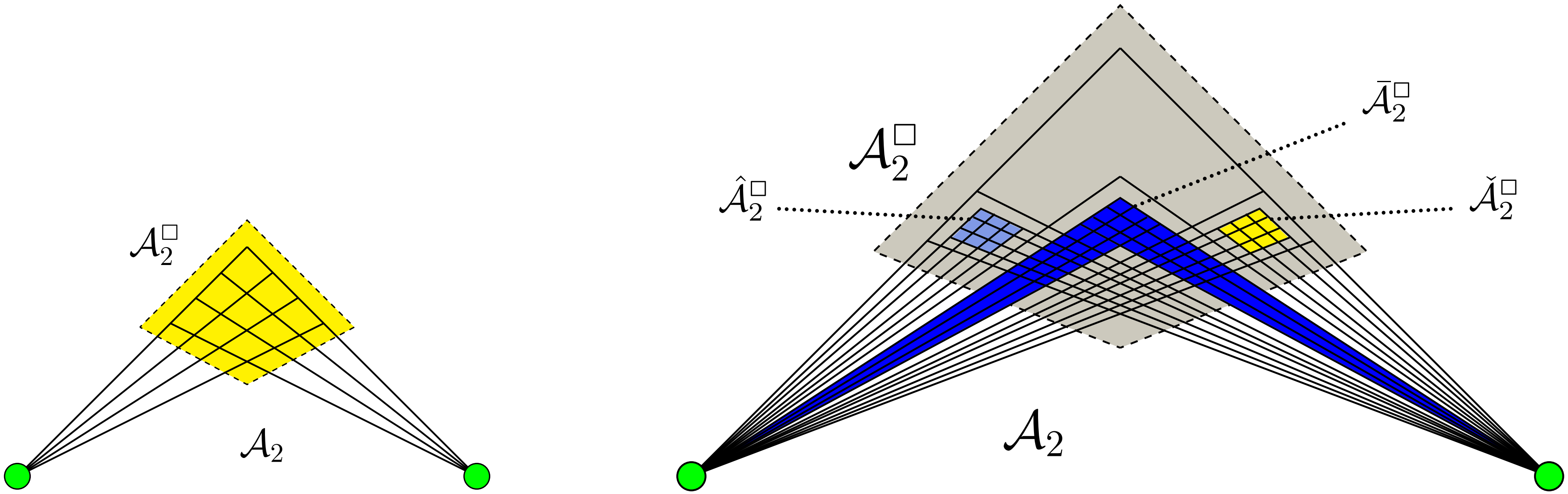
$$s(n, g) \leq \frac{7g}{3} - 1$$

Arrangements of line segments: a lower bound of $\Omega(g \log \frac{n}{g})$



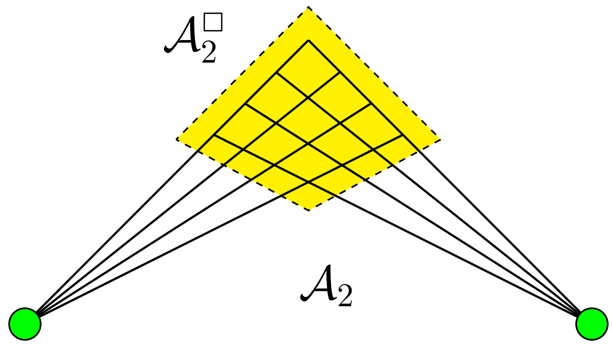
$$s(n, 2) = \Omega(\log n)$$

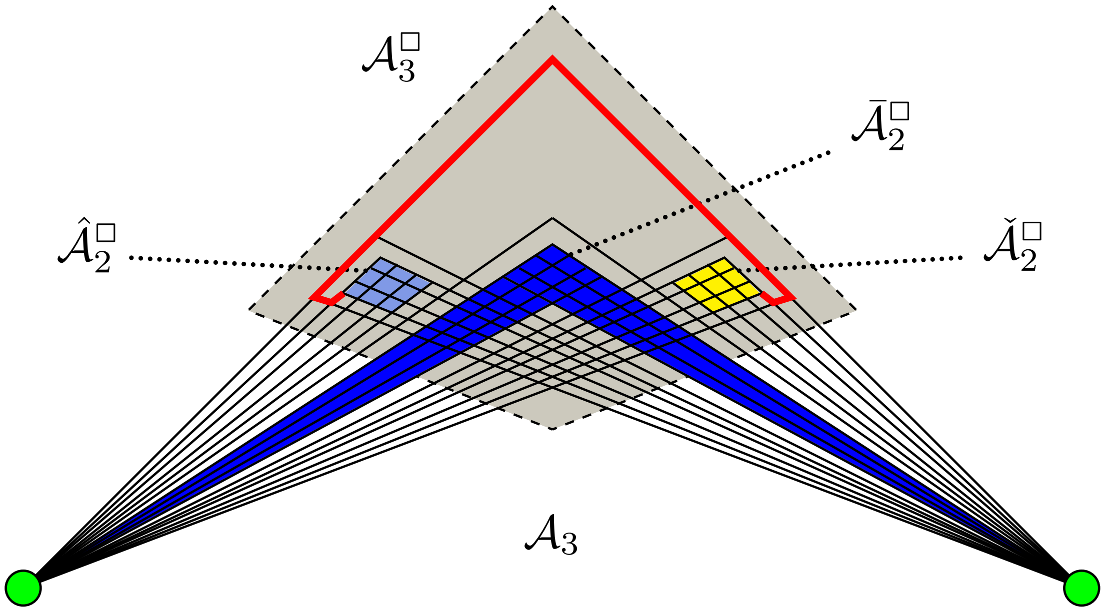
Arrangements of line segments: a lower bound of $\Omega(g \log \frac{n}{g})$



$$s(n, 2) = \Omega(\log n)$$

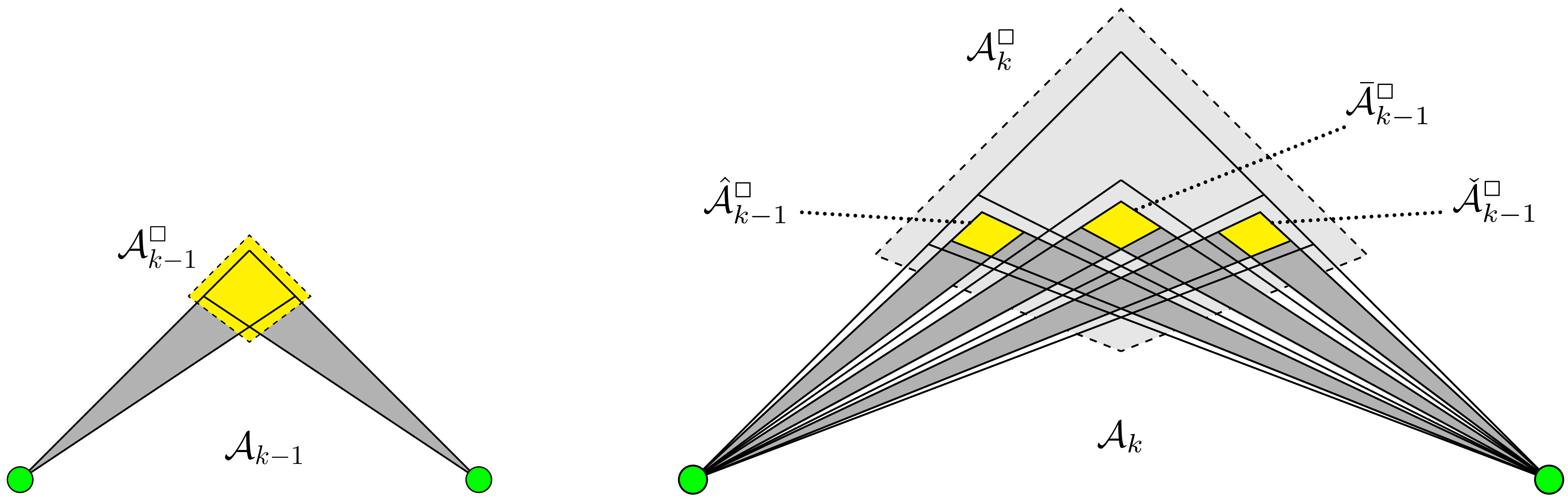
Arrangements of line segments: a lower bound of $\Omega(g \log \frac{n}{g})$





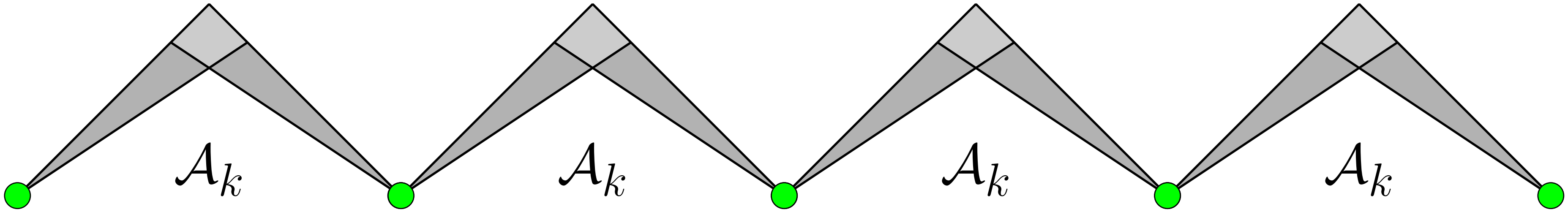
$$s(n, 2) = \Omega(\log n)$$

Arrangements of line segments: a lower bound of $\Omega(g \log \frac{n}{g})$



$$s(n, 2) = \Omega(\log n)$$

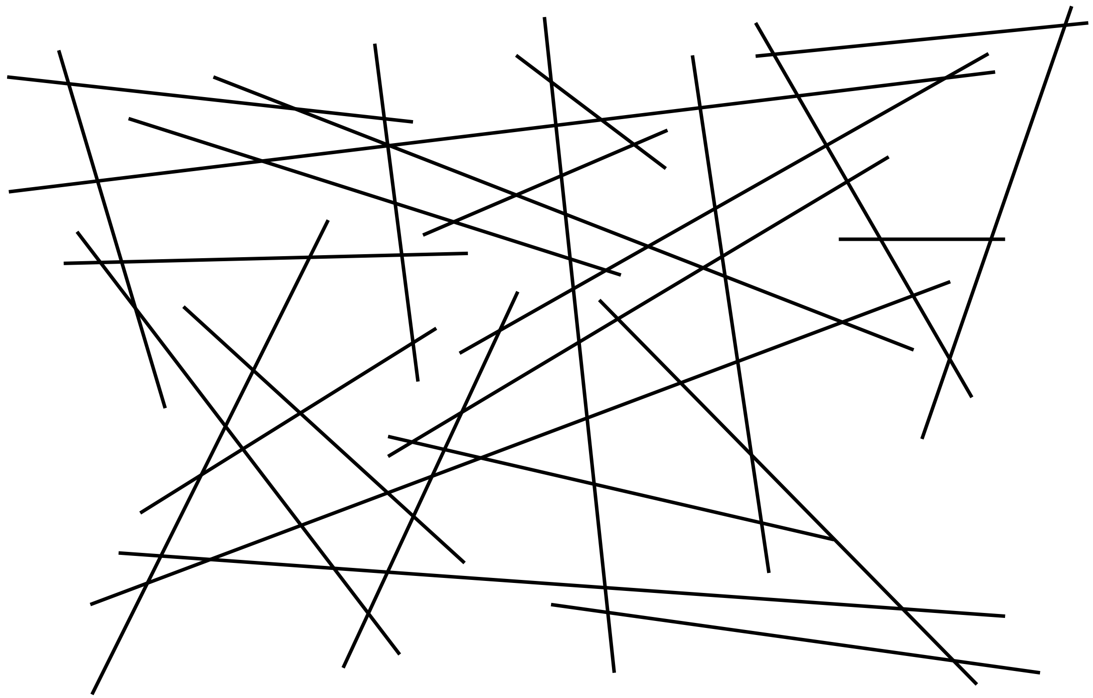
Arrangements of line segments: a lower bound of $\Omega(g \log \frac{n}{g})$



$$s(n, g) = \Omega(g \log \frac{n}{g})$$

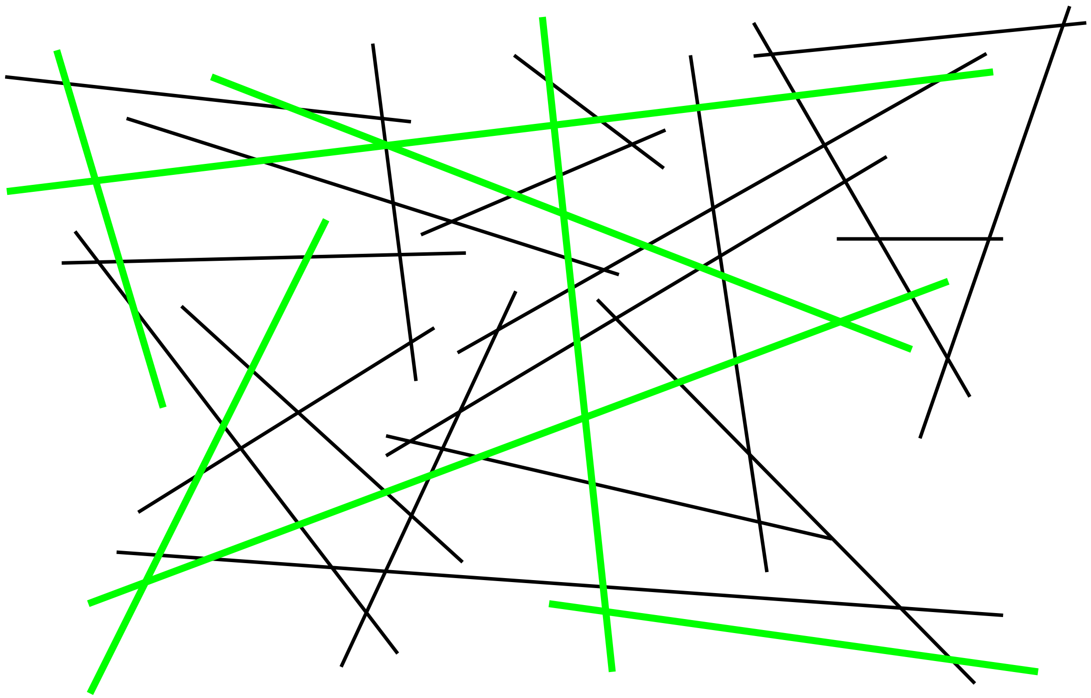
Arrangements of line segments: an upper bound of $O(g^2 \log n)$

- partitioning into nice arrangements: $O(g)$ searchlights (remain fixed)
- recursive searching of nice arrangements (divide-and-conquer)
  - $\rightarrow$ depth of the recursion with respect to a guard v: $O(\log n)$
  - $\rightarrow$ divide-and-conquer: $O(1)$ searchlights per each guard
  - $\rightarrow$ the total number of $O(g^2 \log n)$ searchlights



Arrangements of line segments: an upper bound of $O(g^2 \log n)$

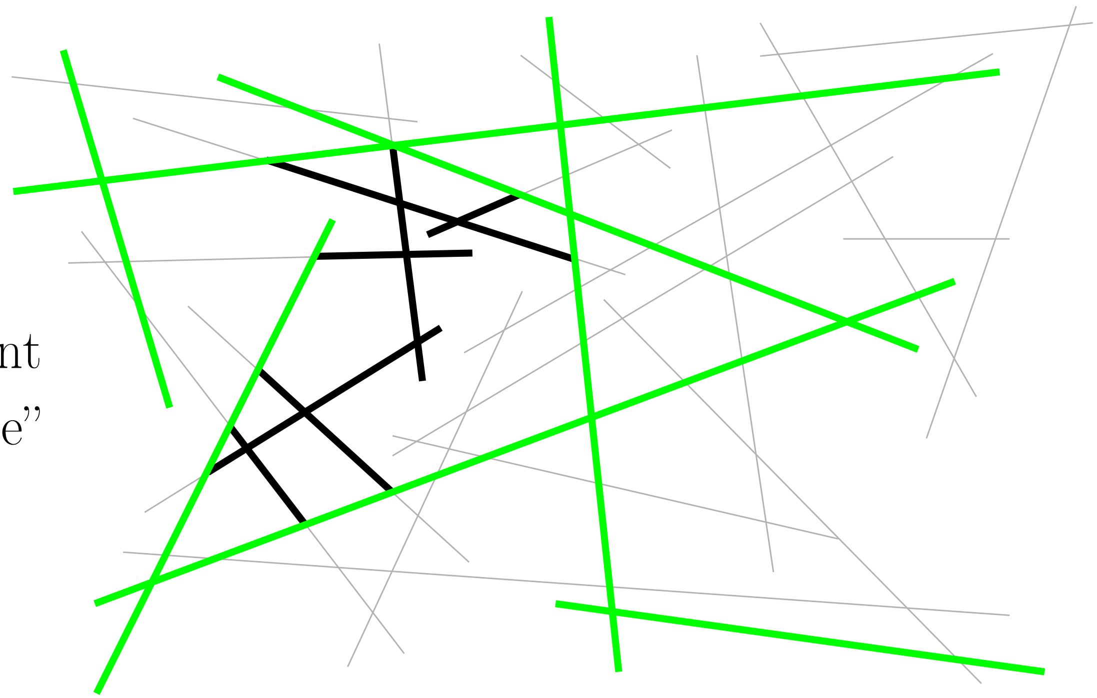
- partitioning into nice arrangements: $O(g)$ searchlights (remain fixed)
- recursive searching of nice arrangements (divide-and-conquer)
  - $\rightarrow$ depth of the recursion with respect to a guard v: $O(\log n)$
  - $\rightarrow$ divide-and-conquer: $O(1)$ searchlights per each guard
  - $\rightarrow$ the total number of $O(g^2 \log n)$ searchlights



Arrangements of line segments: an upper bound of $O(g^2 \log n)$

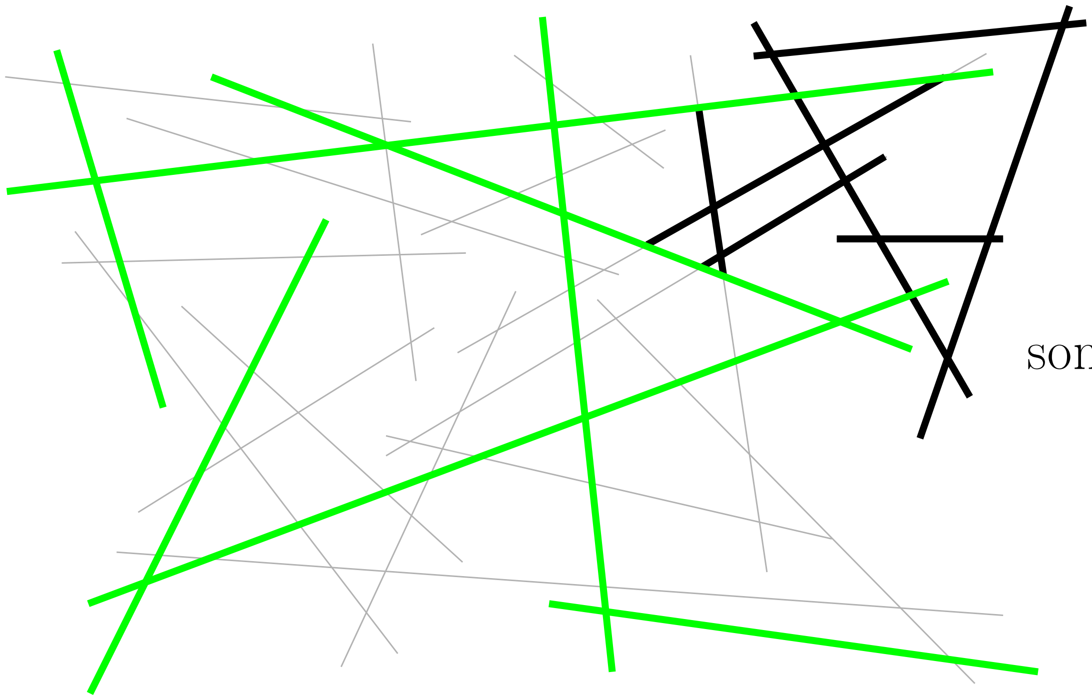
- partitioning into nice arrangements: $O(g)$ searchlights (remain fixed)
- recursive searching of nice arrangements (divide-and-conquer)
  - $\rightarrow$ depth of the recursion with respect to a guard v: $O(\log n)$
  - $\rightarrow$ divide-and-conquer: $O(1)$ searchlights per each guard
  - $\rightarrow$ the total number of $O(g^2 \log n)$ searchlights

nice arrangement
all guards are "outside"



Arrangements of line segments: an upper bound of $O(g^2 \log n)$

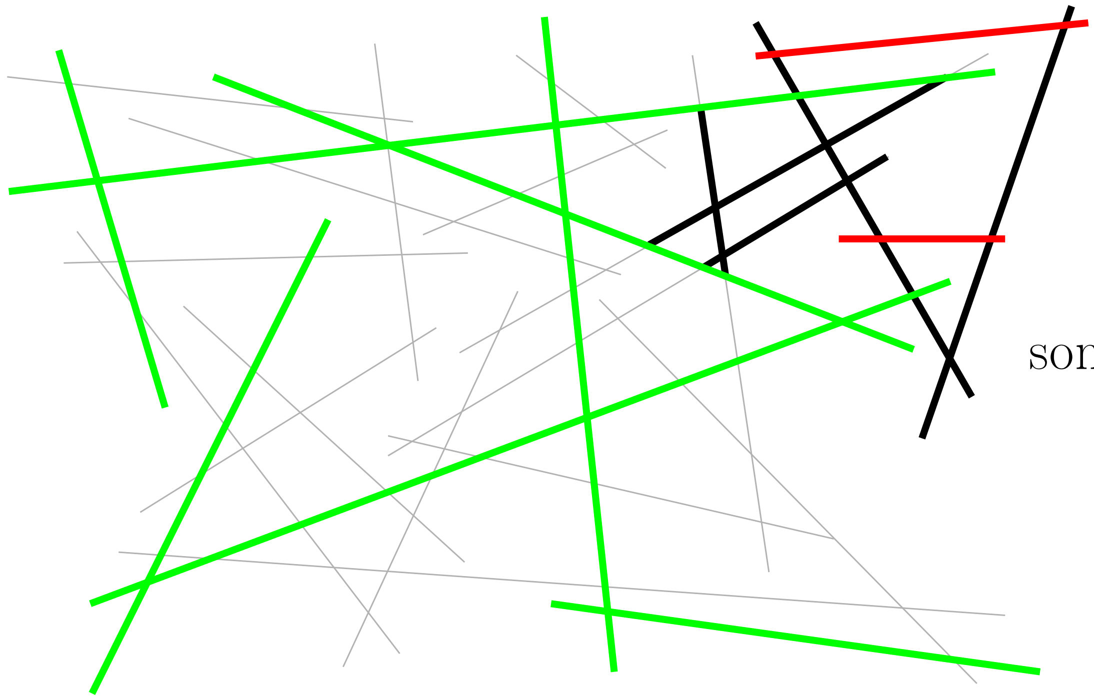
- partitioning into nice arrangements: $O(g)$ searchlights (remain fixed)
- recursive searching of nice arrangements (divide-and-conquer)
  - $\rightarrow$ depth of the recursion with respect to a guard v: $O(\log n)$
  - $\rightarrow$ divide-and-conquer: $O(1)$ searchlights per each guard
  - $\rightarrow$ the total number of $O(g^2 \log n)$ searchlights



not nice arrangement
some guards are "inside"

Arrangements of line segments: an upper bound of $O(g^2 \log n)$

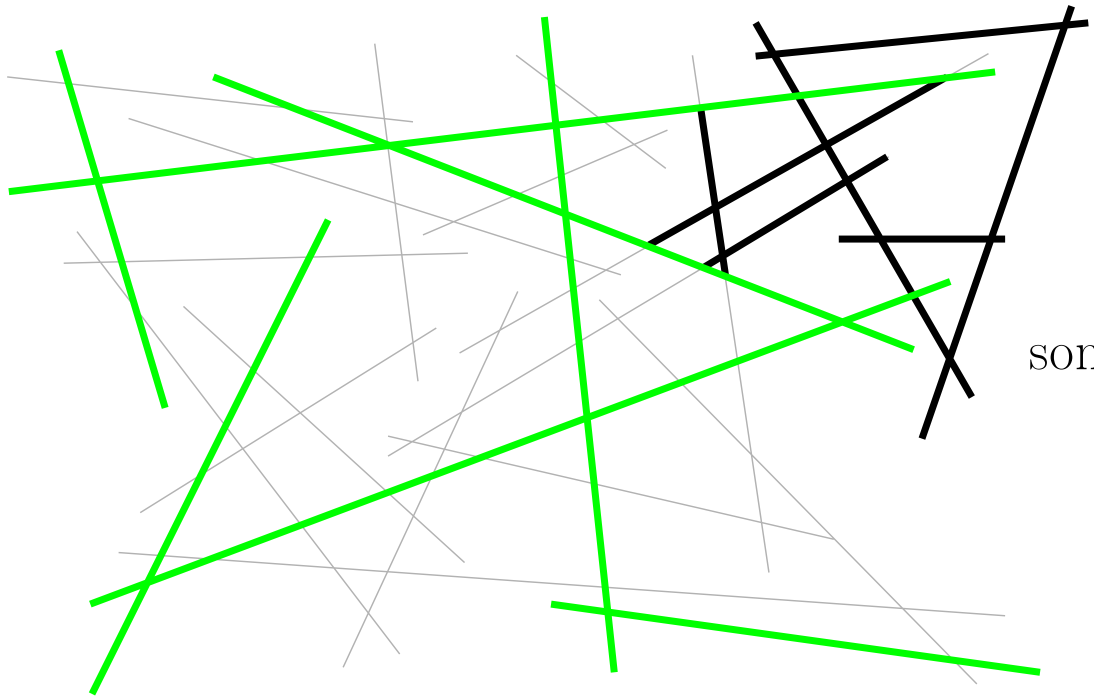
- partitioning into nice arrangements: $O(g)$ searchlights (remain fixed)
- recursive searching of nice arrangements (divide-and-conquer)
  - $\rightarrow$ depth of the recursion with respect to a guard v: $O(\log n)$
  - $\rightarrow$ divide-and-conquer: $O(1)$ searchlights per each guard
  - $\rightarrow$ the total number of $O(g^2 \log n)$ searchlights



not nice arrangement
some guards are "inside"

Arrangements of line segments: an upper bound of $O(g^2 \log n)$

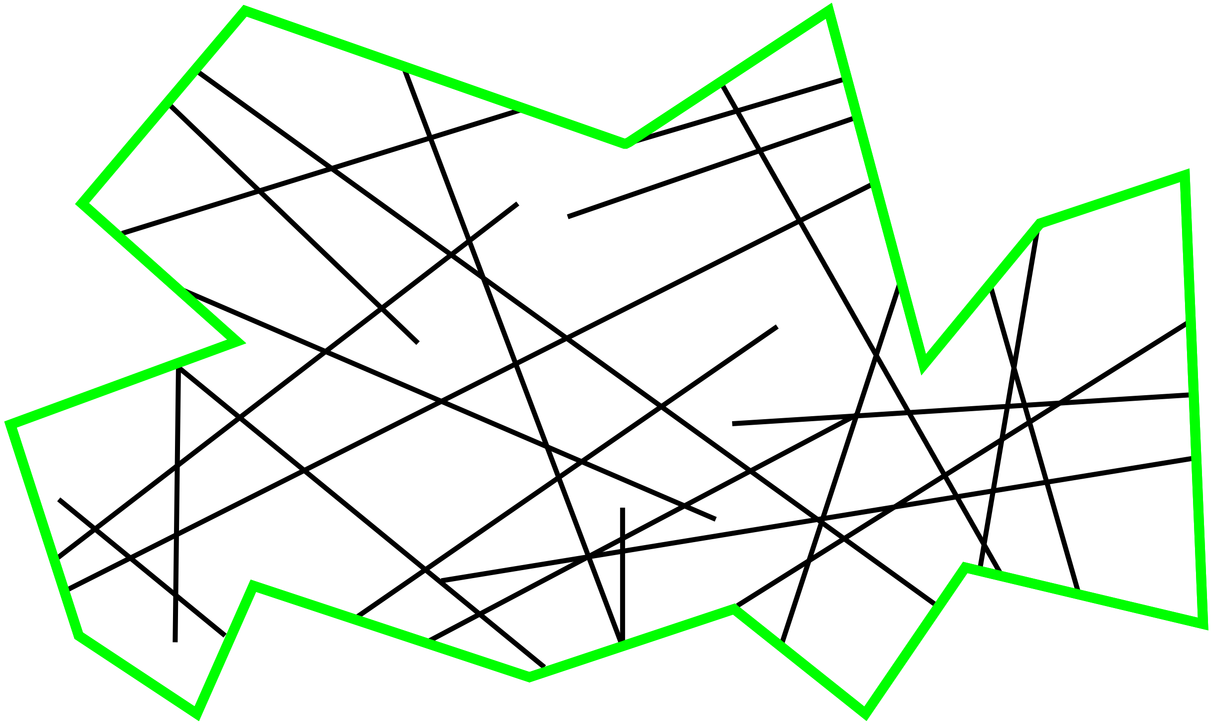
- partitioning into nice arrangements: $O(g)$ searchlights (remain fixed)
- recursive searching of nice arrangements (divide-and-conquer)
  - $\rightarrow$ depth of the recursion with respect to a guard v: $O(\log n)$
  - $\rightarrow$ divide-and-conquer: $O(1)$ searchlights per each guard
  - $\rightarrow$ the total number of $O(g^2 \log n)$ searchlights



not nice arrangement
some guards are "inside"

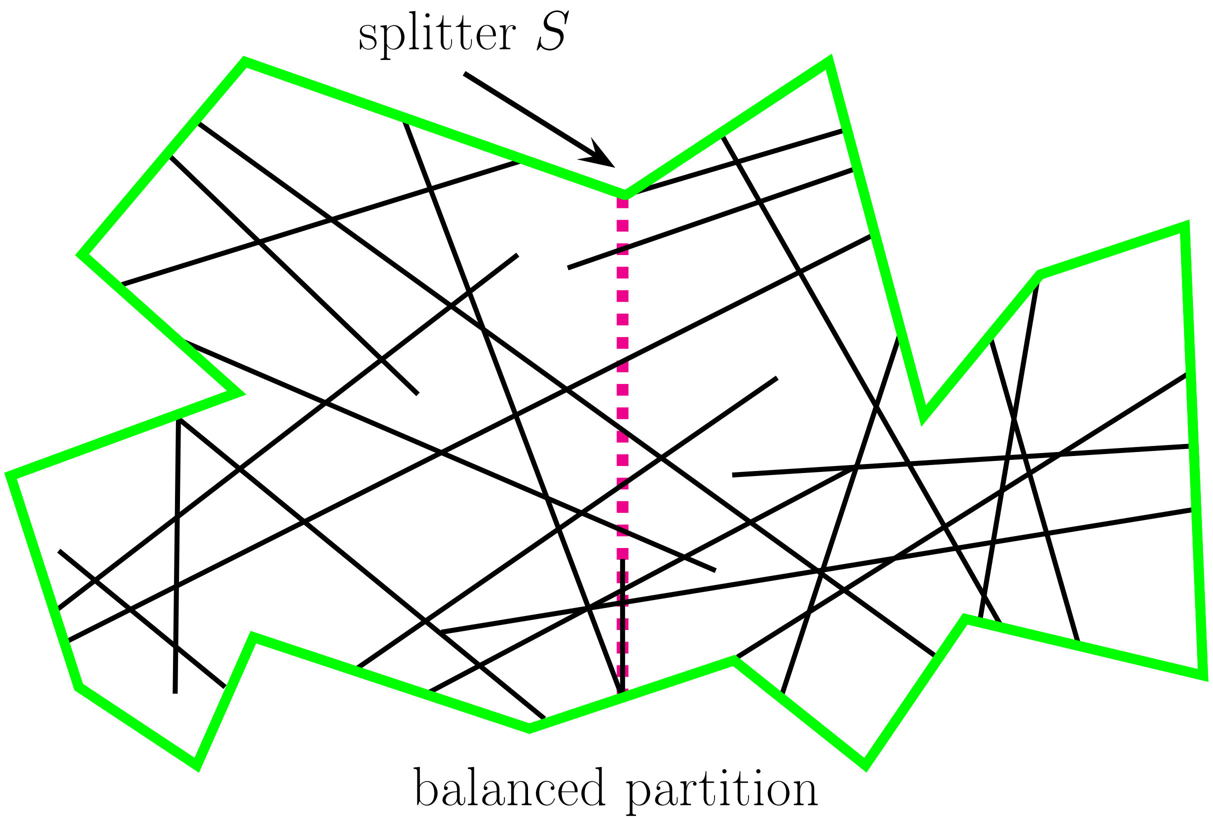
Arrangements of line segments: an upper bound of $O(g^2 \log n)$

- partitioning into nice arrangements: $O(g)$ searchlights (remain fixed)
- recursive searching of nice arrangements (divide-and-conquer)
  - $\rightarrow$ depth of the recursion with respect to a guard v: $O(\log n)$
  - $\rightarrow$ divide-and-conquer: $O(1)$ searchlights per each guard
  - $\rightarrow$ the total number of $O(g^2 \log n)$ searchlights



Arrangements of line segments: an upper bound of $O(g^2 \log n)$

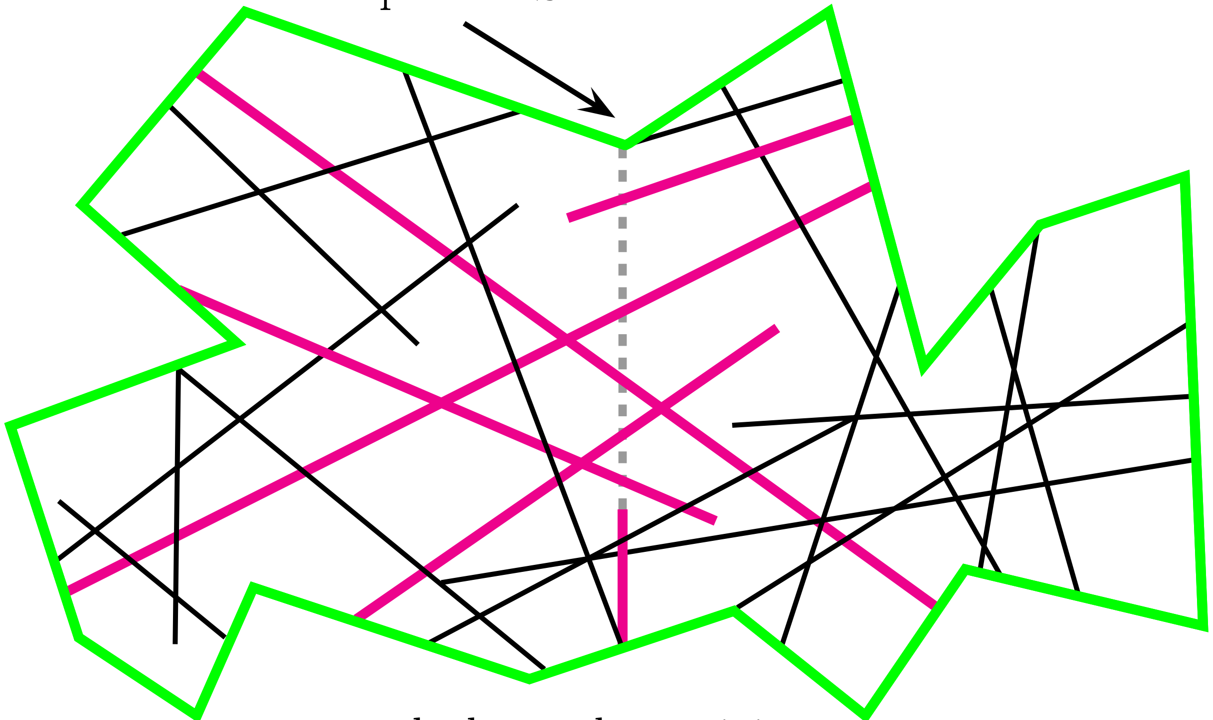
- partitioning into nice arrangements: $O(g)$ searchlights (remain fixed)
- recursive searching of nice arrangements (divide-and-conquer)
  - $\rightarrow$ depth of the recursion with respect to a guard v: $O(\log n)$
  - $\rightarrow$ divide-and-conquer: $O(1)$ searchlights per each guard
  - $\rightarrow$ the total number of $O(g^2 \log n)$ searchlights



Arrangements of line segments: an upper bound of $O(g^2 \log n)$

- partitioning into nice arrangements: $O(g)$ searchlights (remain fixed)
- recursive searching of nice arrangements (divide-and-conquer)
  - $\rightarrow$ depth of the recursion with respect to a guard v: $O(\log n)$
  - $\rightarrow$ divide-and-conquer: $O(1)$ searchlights per each guard
  - $\rightarrow$ the total number of $O(g^2 \log n)$ searchlights

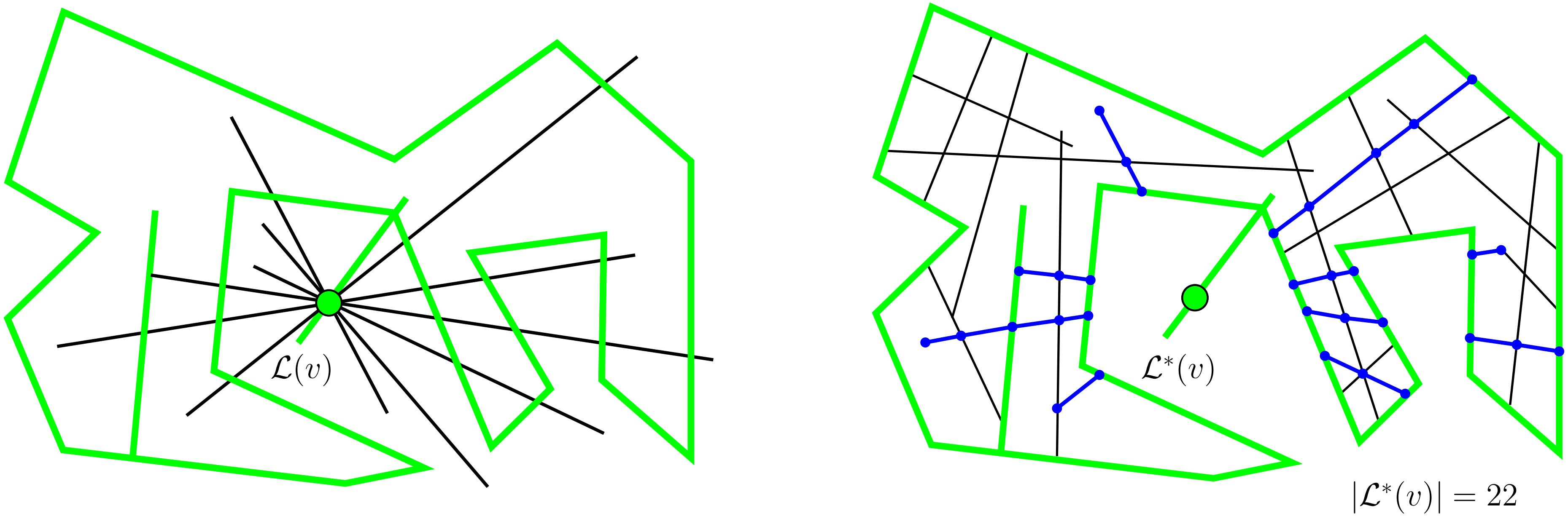
splitter S



balanced partition

Arrangements of line segments: an upper bound of $O(g^2 \log n)$

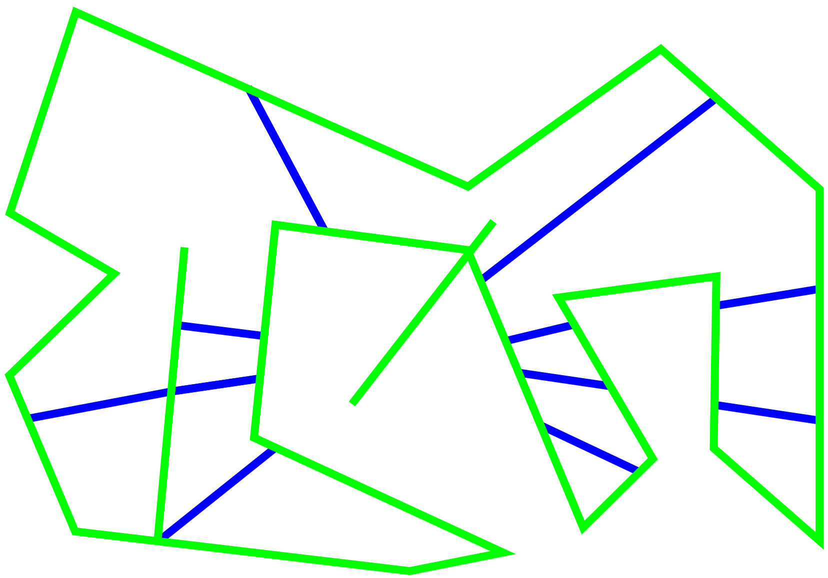
- partitioning into nice arrangements: $O(g)$ searchlights (remain fixed)
- recursive searching of nice arrangements (divide-and-conquer)
  - $\rightarrow$ depth of the recursion with respect to a guard v: $O(\log n)$
  - $\rightarrow$ divide-and-conquer: $O(1)$ searchlights per each guard
  - $\rightarrow$ the total number of $O(g^2 \log n)$ searchlights

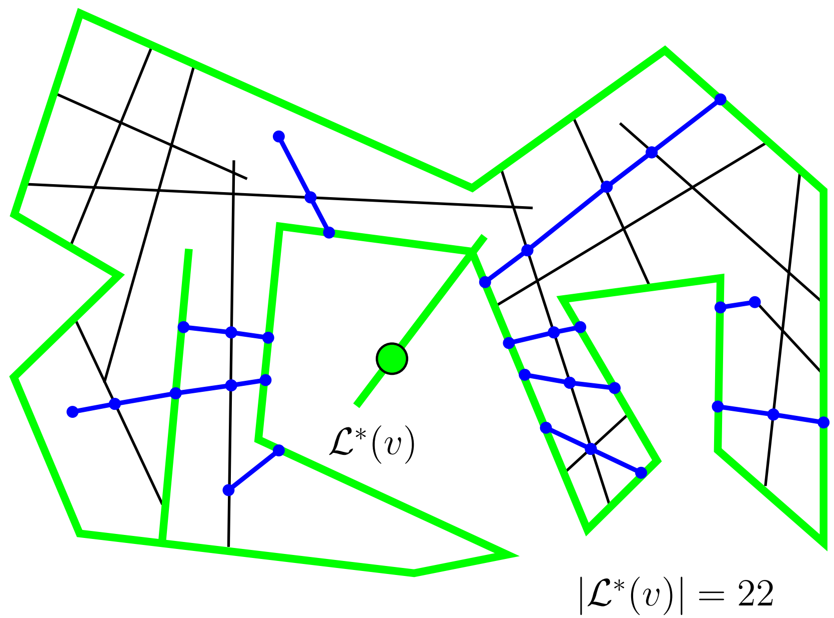


finding a balanced splitter with respect to v

Arrangements of line segments: an upper bound of $O(g^2 \log n)$

- partitioning into nice arrangements: $O(g)$ searchlights (remain fixed)
- recursive searching of nice arrangements (divide-and-conquer)
  - $\to$ depth of the recursion with respect to a guard v: $O(\log n)$
  - $\to$ divide-and-conquer: $O(1)$ searchlights per each guard
  - $\to$ the total number of $O(g^2 \log n)$ searchlights

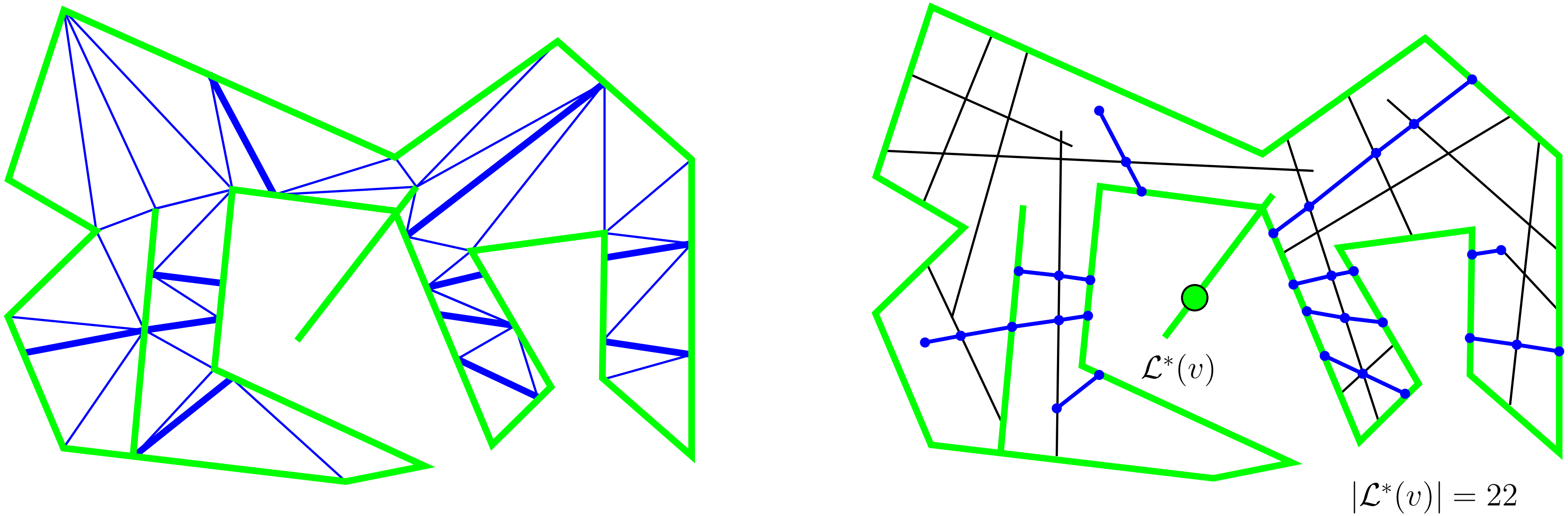




finding a balanced splitter with respect to v

Arrangements of line segments: an upper bound of $O(g^2 \log n)$

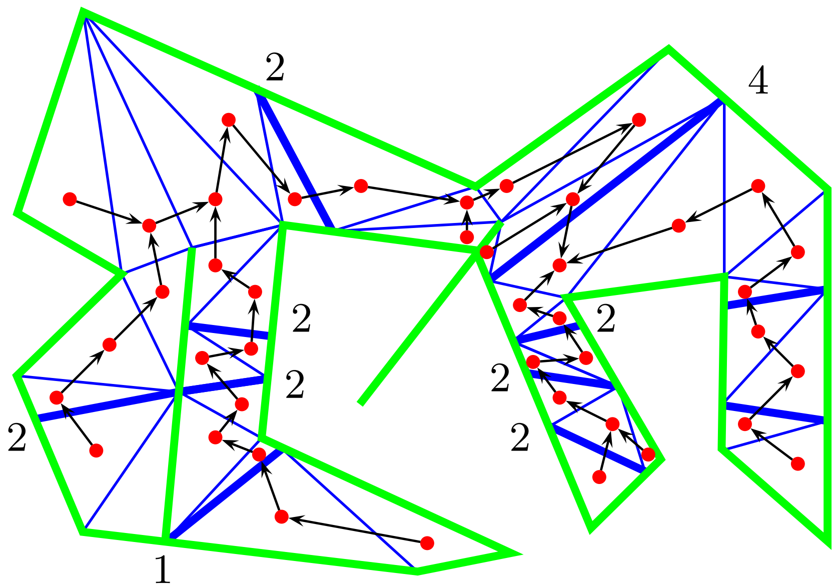
- partitioning into nice arrangements: $O(g)$ searchlights (remain fixed)
- recursive searching of nice arrangements (divide-and-conquer)
  - $\rightarrow$ depth of the recursion with respect to a guard v: $O(\log n)$
  - $\rightarrow$ divide-and-conquer: $O(1)$ searchlights per each guard
  - $\rightarrow$ the total number of $O(g^2 \log n)$ searchlights

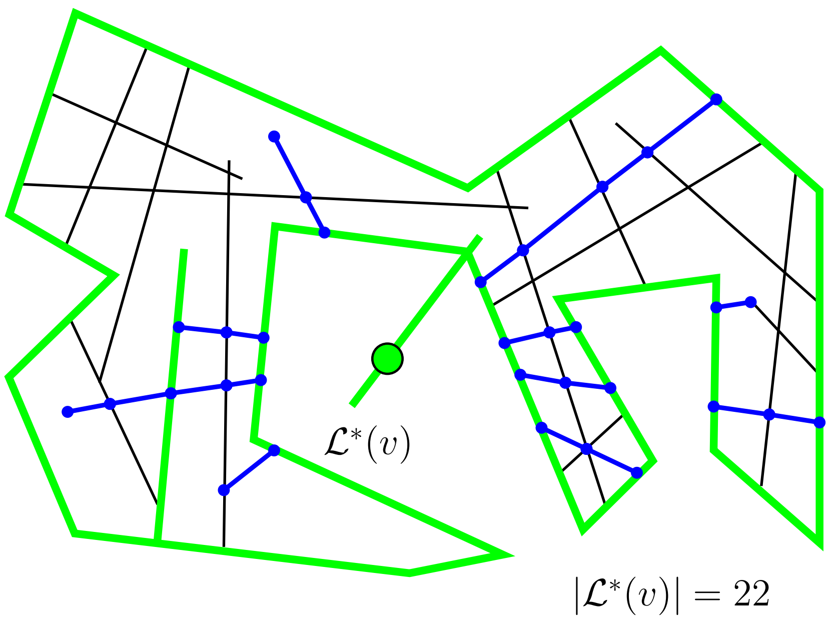


finding a balanced splitter with respect to v

Arrangements of line segments: an upper bound of $O(g^2 \log n)$

- partitioning into nice arrangements: $O(g)$ searchlights (remain fixed)
- recursive searching of nice arrangements (divide-and-conquer)
  - $\rightarrow$ depth of the recursion with respect to a guard v: $O(\log n)$
  - $\rightarrow$ divide-and-conquer: $O(1)$ searchlights per each guard
  - $\rightarrow$ the total number of $O(g^2 \log n)$ searchlights

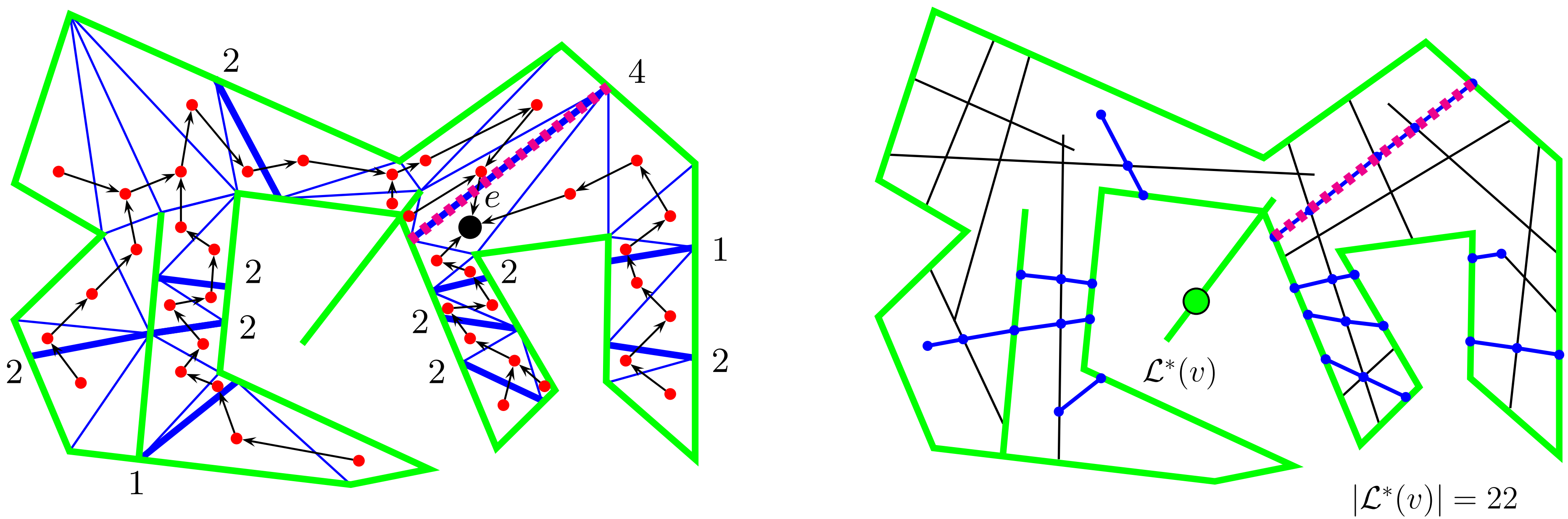




finding a balanced splitter with respect to v

Arrangements of line segments: an upper bound of $O(g^2 \log n)$

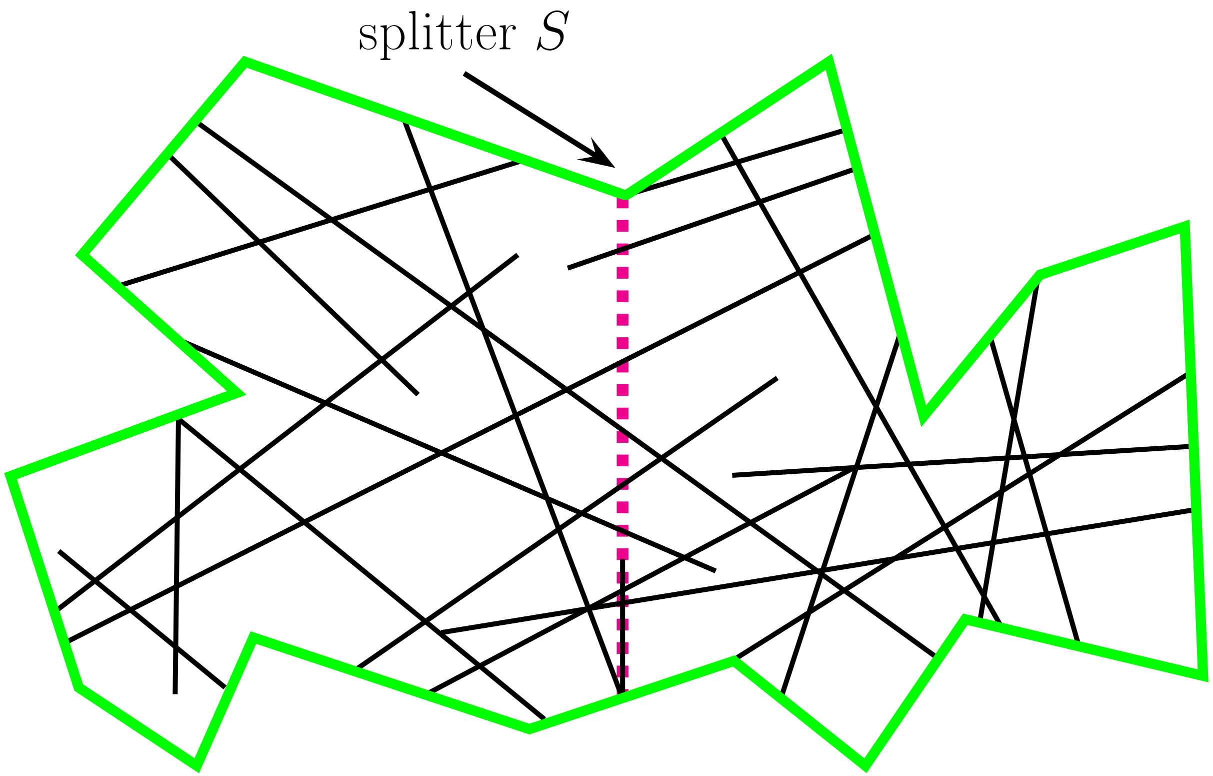
- partitioning into nice arrangements: $O(g)$ searchlights (remain fixed)
- recursive searching of nice arrangements (divide-and-conquer)
  - $\rightarrow$ depth of the recursion with respect to a guard v: $O(\log n)$
  - $\rightarrow$ divide-and-conquer: $O(1)$ searchlights per each guard
  - $\rightarrow$ the total number of $O(g^2 \log n)$ searchlights



finding a balanced splitter with respect to v

Arrangements of line segments: an upper bound of $O(g^2 \log n)$

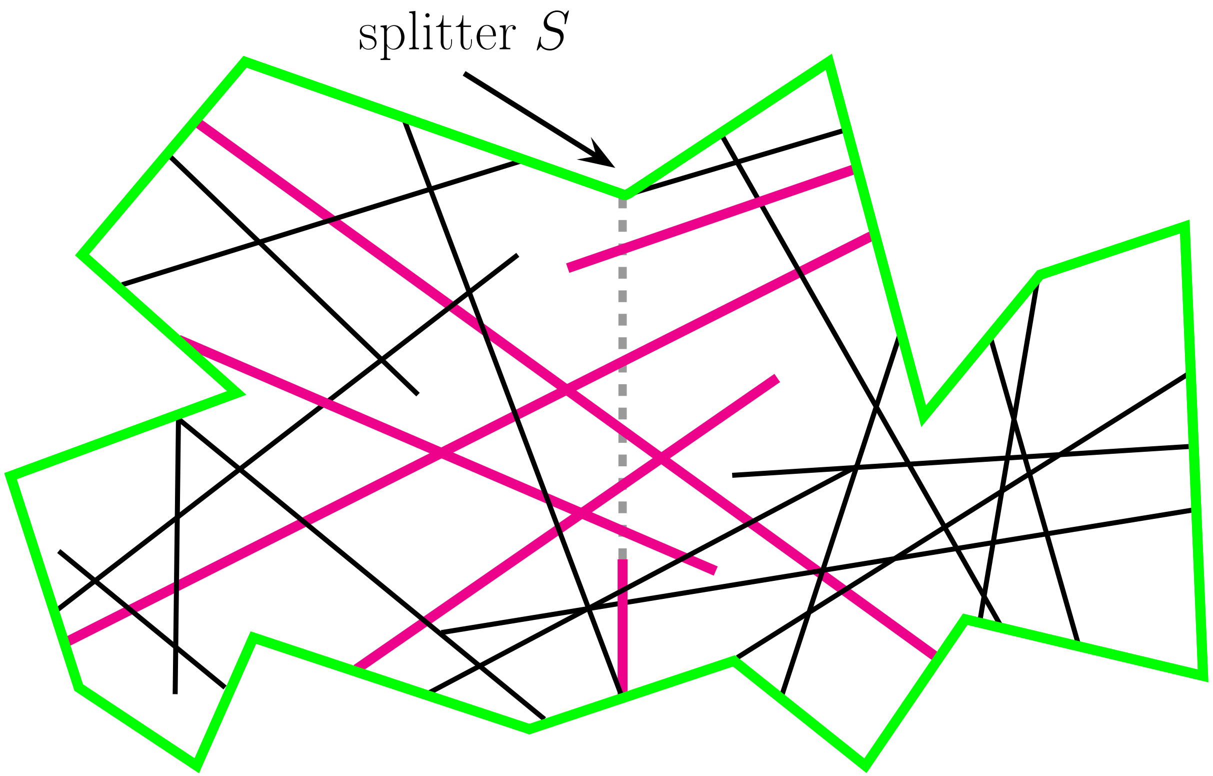
- partitioning into nice arrangements: $O(g)$ searchlights (remain fixed)
- recursive searching of nice arrangements (divide-and-conquer)
  - $\rightarrow$ depth of the recursion with respect to a guard v: $O(\log n)$
  - $\rightarrow$ divide-and-conquer: $O(1)$ searchlights per each guard
  - $\rightarrow$ the total number of $O(g^2 \log n)$ searchlights



balanced partition with respect to v

Arrangements of line segments: an upper bound of $O(g^2 \log n)$

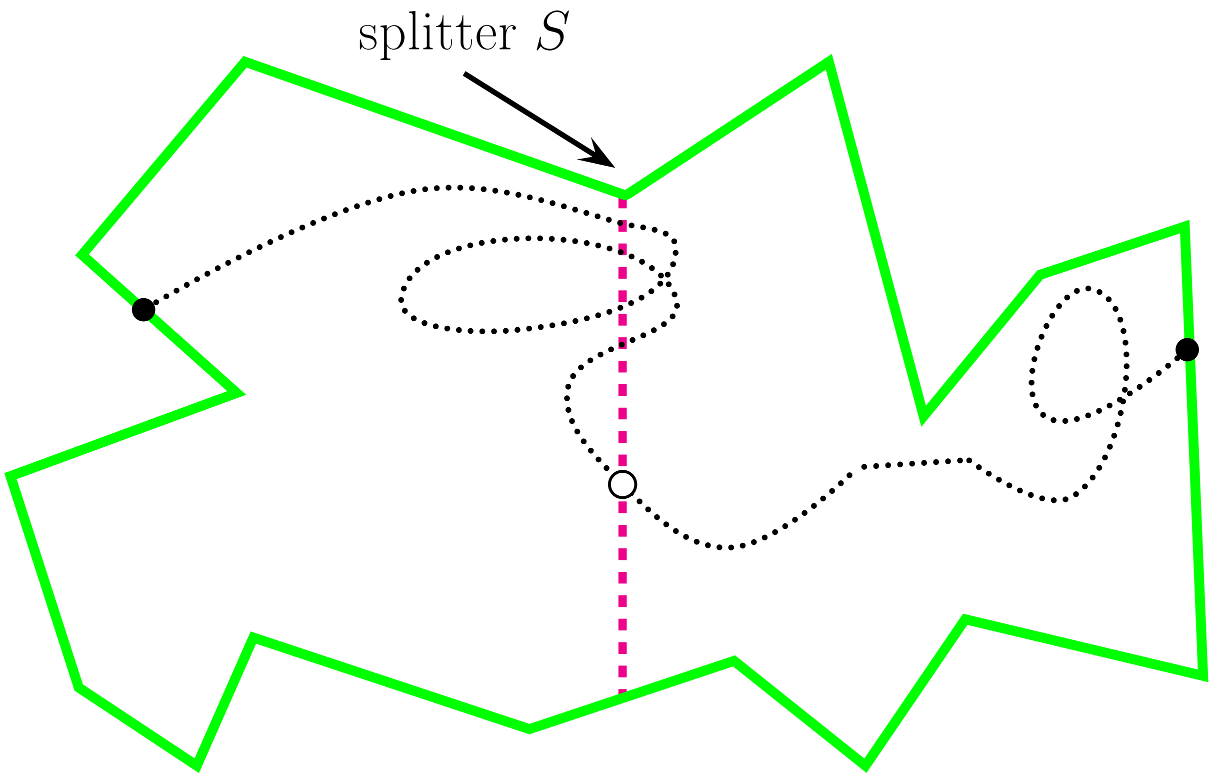
- partitioning into nice arrangements: $O(g)$ searchlights (remain fixed)
- recursive searching of nice arrangements (divide-and-conquer)
  - $\rightarrow$ depth of the recursion with respect to a guard v: $O(\log n)$
  - $\rightarrow$ divide-and-conquer: $O(1)$ searchlights per each guard
  - $\rightarrow$ the total number of $O(g^2 \log n)$ searchlights



balanced partition with respect to v

Arrangements of line segments: an upper bound of $O(g^2 \log n)$

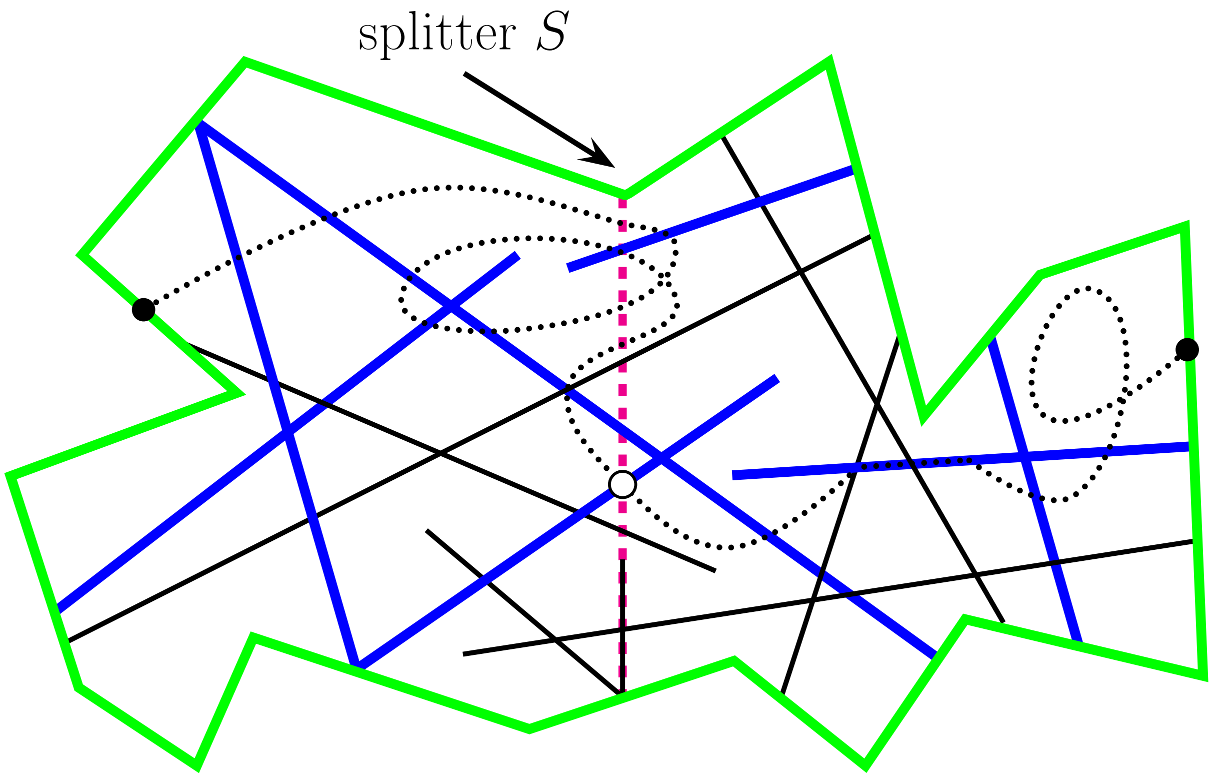
- partitioning into nice arrangements: $O(g)$ searchlights (remain fixed)
- recursive searching of nice arrangements (divide-and-conquer)
  - $\rightarrow$ depth of the recursion with respect to a guard v: $O(\log n)$
  - $\rightarrow$ divide-and-conquer: $O(1)$ searchlights per each guard
  - $\rightarrow$ the total number of $O(g^2 \log n)$ searchlights



curve $\mathcal{C}$ and its (S, x)-separator

Arrangements of line segments: an upper bound of $O(g^2 \log n)$

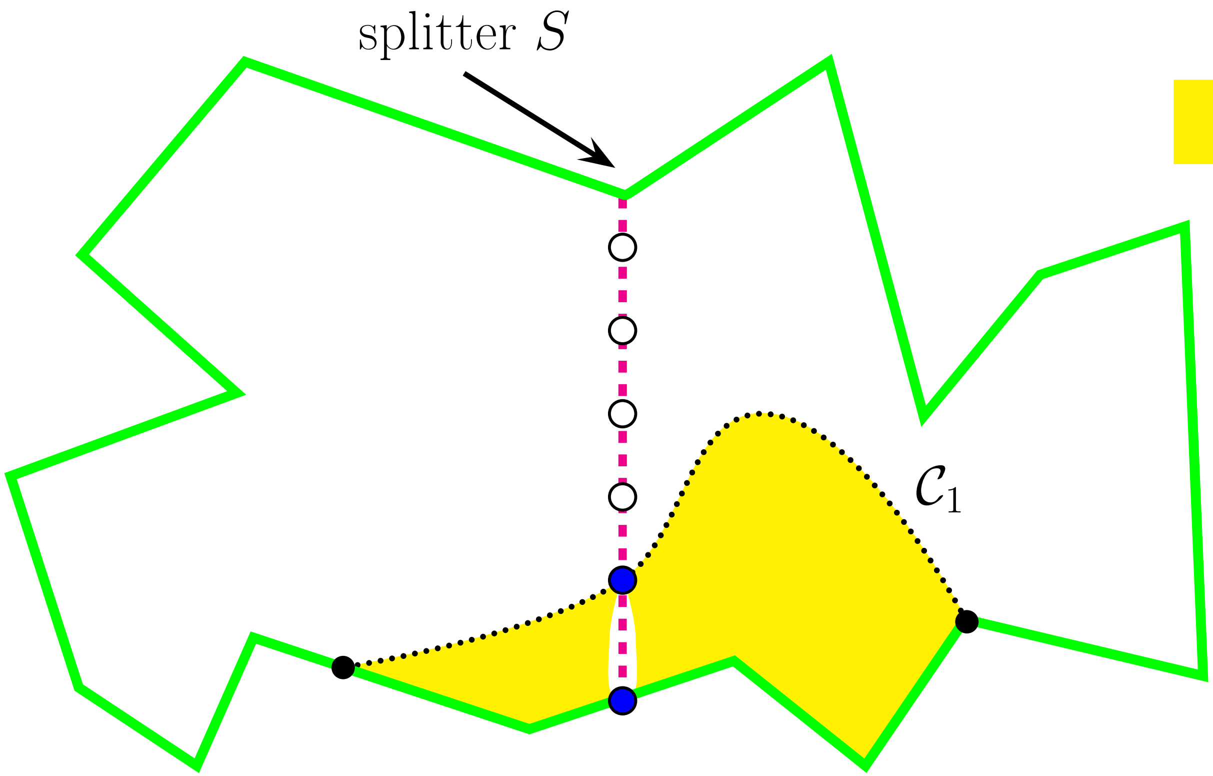
- partitioning into nice arrangements: $O(g)$ searchlights (remain fixed)
- recursive searching of nice arrangements (divide-and-conquer)
  - $\rightarrow$ depth of the recursion with respect to a guard v: $O(\log n)$
  - $\rightarrow$ divide-and-conquer: $O(1)$ searchlights per each guard
  - $\rightarrow$ the total number of $O(g^2 \log n)$ searchlights



curve $\mathcal{C}$ and its (S, x)-separator

Arrangements of line segments: an upper bound of $O(g^2 \log n)$

- partitioning into nice arrangements: $O(g)$ searchlights (remain fixed)
- recursive searching of nice arrangements (divide-and-conquer)
  - $\rightarrow$ depth of the recursion with respect to a guard v: $O(\log n)$
  - $\rightarrow$ divide-and-conquer: $O(1)$ searchlights per each guard
  - $\rightarrow$ the total number of $O(g^2 \log n)$ searchlights



recursive clearing **balanced** arrangements along the splitter

recursion

$O(1)$ searchlights for "illuminating" $\mathcal{C}_1$

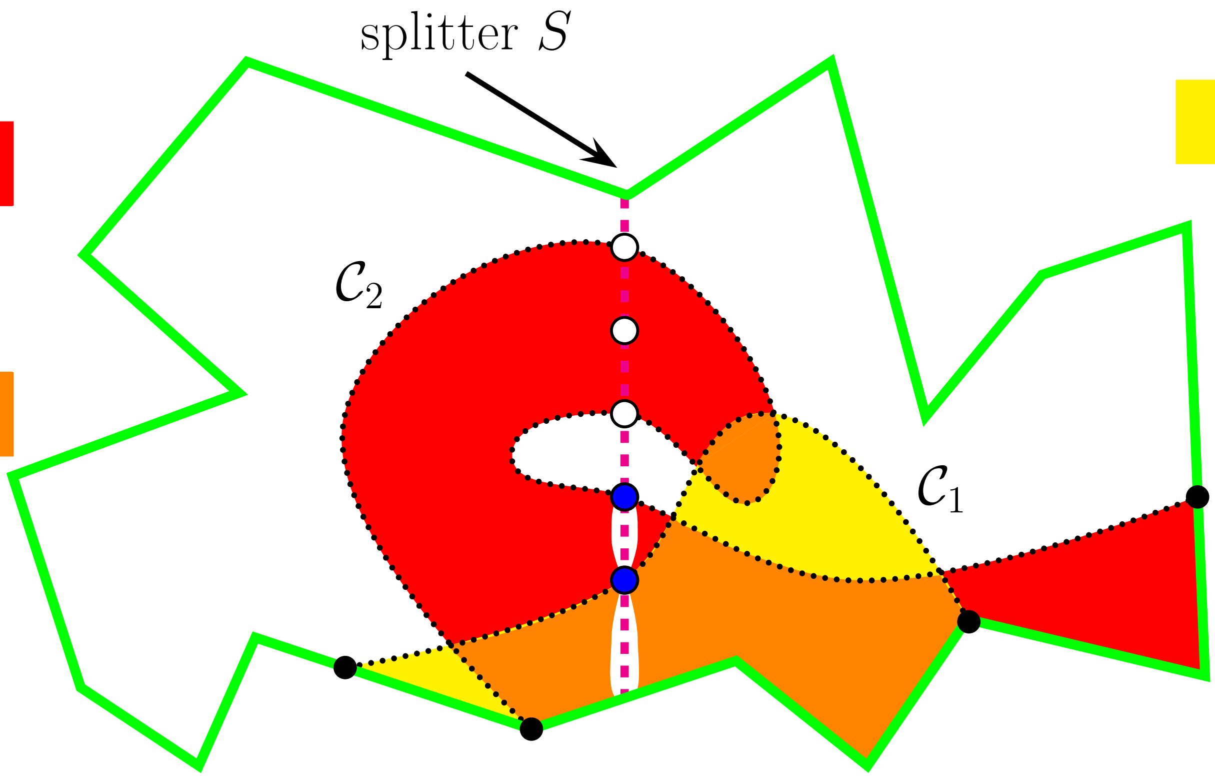
Arrangements of line segments: an upper bound of $O(g^2 \log n)$

- partitioning into nice arrangements: $O(g)$ searchlights (remain fixed)
- recursive searching of nice arrangements (divide-and-conquer)
  - $\rightarrow$ depth of the recursion with respect to a guard v: $O(\log n)$
  - $\rightarrow$ divide-and-conquer: $O(1)$ searchlights per each guard
  - $\rightarrow$ the total number of $O(g^2 \log n)$ searchlights

recursion

$O(1)$ searchlights for "illuminating" $\mathcal{C}_2$

searched



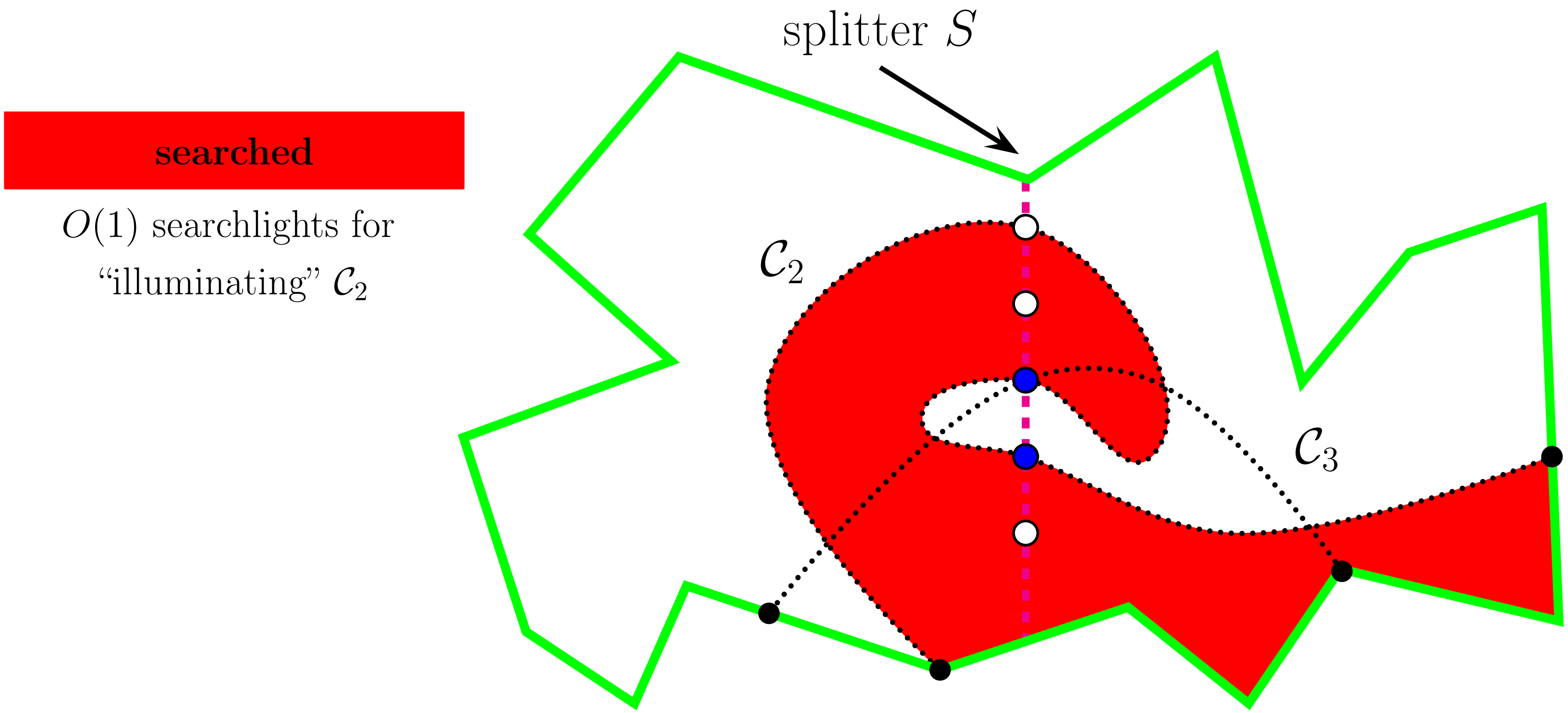
searched

$O(1)$ searchlights for "illuminating" $\mathcal{C}_1$

recursive clearing **balanced** arrangements along the splitter

Arrangements of line segments: an upper bound of $O(g^2 \log n)$

- partitioning into nice arrangements: $O(g)$ searchlights (remain fixed)
- recursive searching of nice arrangements (divide-and-conquer)
  - $\rightarrow$ depth of the recursion with respect to a guard v: $O(\log n)$
  - $\rightarrow$ divide-and-conquer: $O(1)$ searchlights per each guard
  - $\rightarrow$ the total number of $O(g^2 \log n)$ searchlights



recursive clearing **balanced** arrangements along the splitter

Problem 1. Provide better estimates on $s(n, g)$ in the case of (n, g)-arrangements of lines. In particular, prove or disprove that $s(n, g) \leq 2g$.

$$2g - 1 \leq s(n, g) \leq \frac{7g}{3} - 1$$

Problem 2. Provide better estimates on $s(n, g)$ in the general case of (n, g)-arrangements of line segments.

Without any strong evidence, we conjecture that the upper bound on $s(n, g)$ can be improved up to $O(g \log \Delta)$, where Δ is the maximum number of maximal line segments of an arrangement having a point in common.

$$s(n, g) = \Omega(g \log \Delta) \quad \text{and} \quad s(n, g) = O(g^2 \log n)$$

Problem 3. The time and space complexity of deciding whether the given arrangement (of lines/line segments) can be searched using $k \geq 1$ searchlights.