From binary search through games to graphs

Przemysław Gordinowicz

Institute of Mathematics
Lodz University of Technology,
Łódź, Poland

11^{th} Workshop on GRAph Searching,
Theory and Applications
Bertinoro October 26, 2023

Co-authors

1. Dariusz Dereniowski (Gdańsk), Karolina Wróbel (Łódź)
2. Dariusz Dereniowski (Gdańsk), Paweł Prałat (Toronto)

Co-authors

1. Dariusz Dereniowski (Gdańsk), Karolina Wróbel (Łódź)
2. Dariusz Dereniowski (Gdańsk), Paweł Prałat (Toronto)

Outline

1. (A little dubious) motivation via multi-criteria optimisation
2. Multidimensional binary search as a game
3. Switch to graphs: edge and pair queries

Outline

1. (A little dubious) motivation via multi-criteria optimisation
2. Multidimensional binary search as a game
3. Switch to graphs: edge and pair queries

Let us start with a definition!

Let us start with a definition!

Definition (Renyi)

Mathematician is a device that turns coffee into theorems.

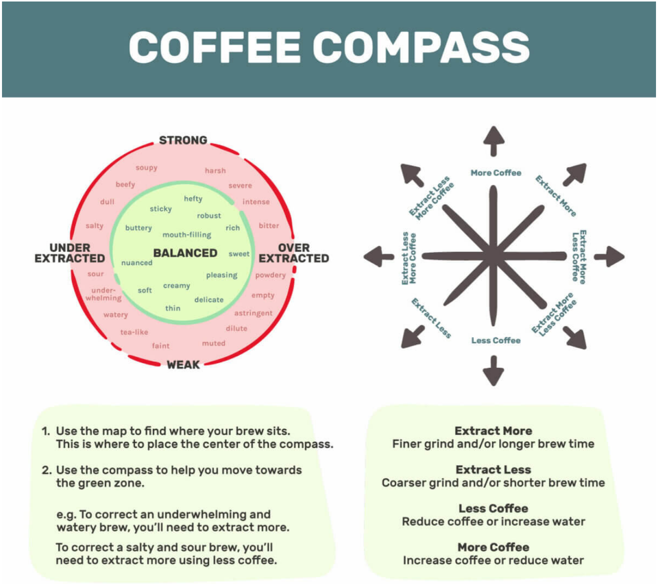
Let us start with a definition!

Definition (Renyi)

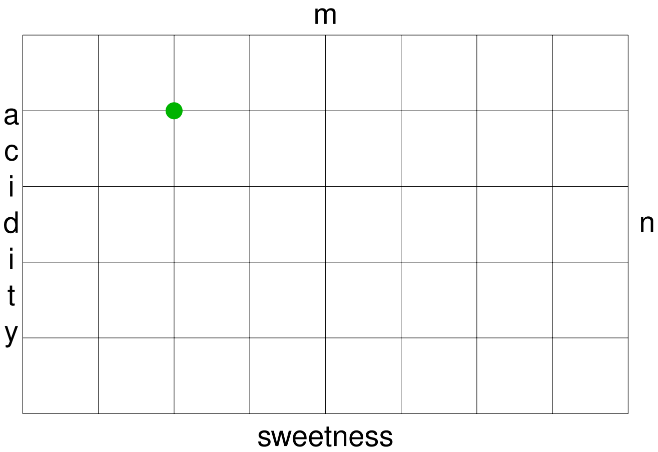
Mathematician is a device that turns coffee into theorems.

Hence, we need a good coffee.
(Computer Scientists too).

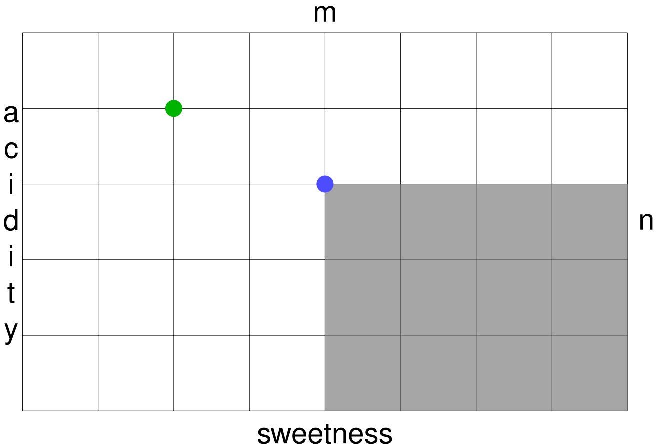
Recipe for good coffee



Let’s try something simpler — maybe lemonade?

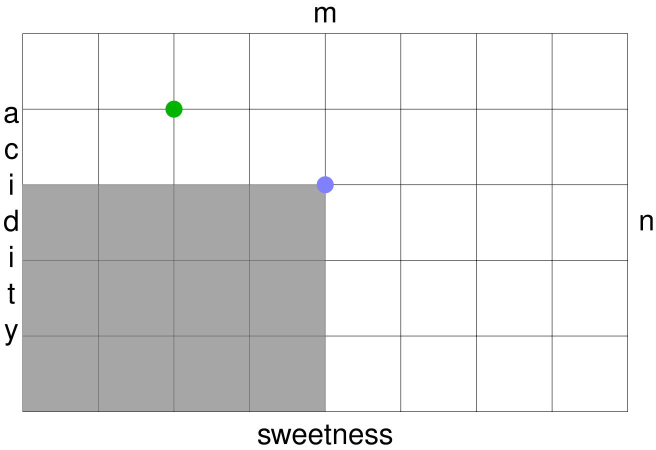


Let's try something simpler — maybe lemonade?

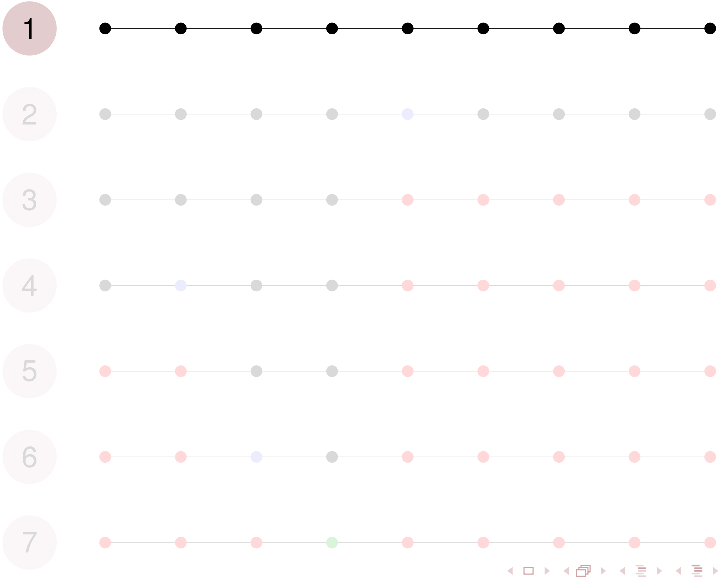


Let’s try something simpler — maybe lemonade?

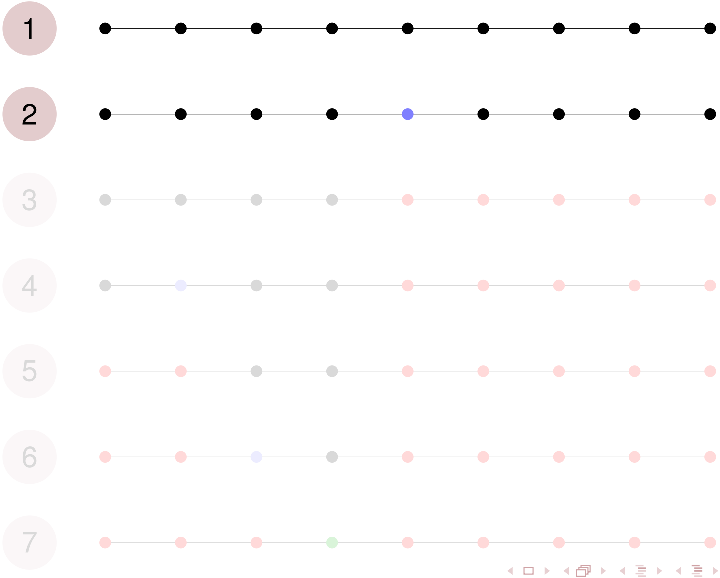
Let's try something simpler — maybe lemonade?



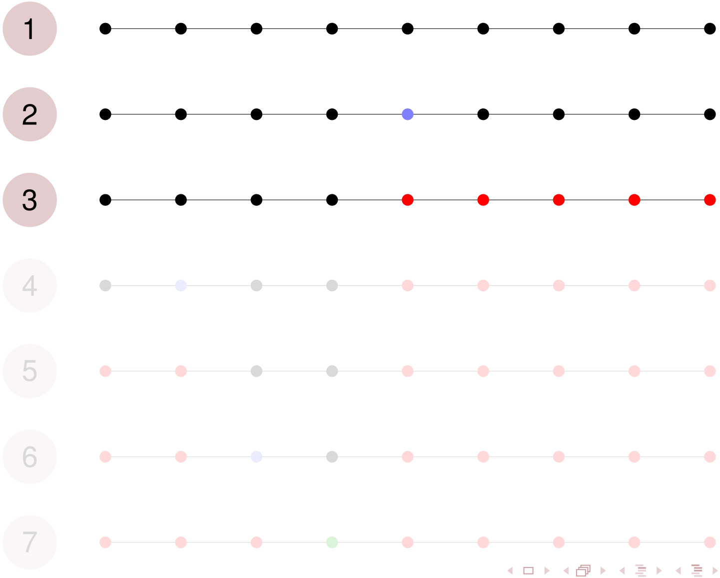
One dimensional solution — a binary search



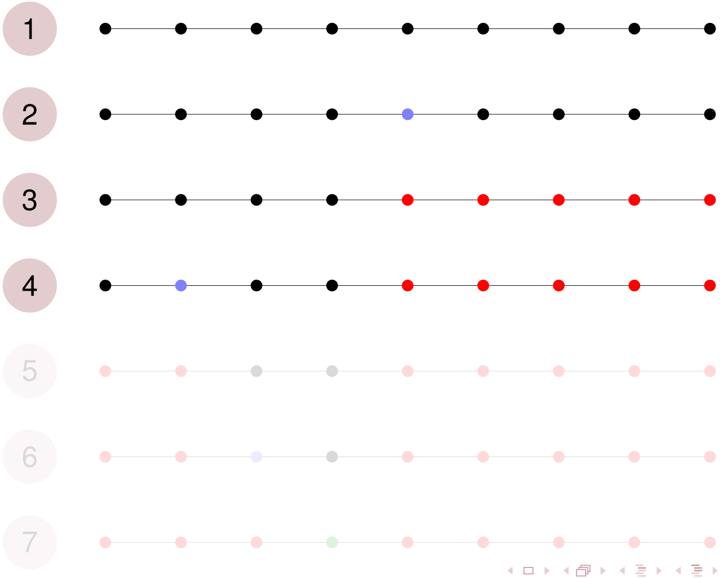
One dimensional solution — a binary search



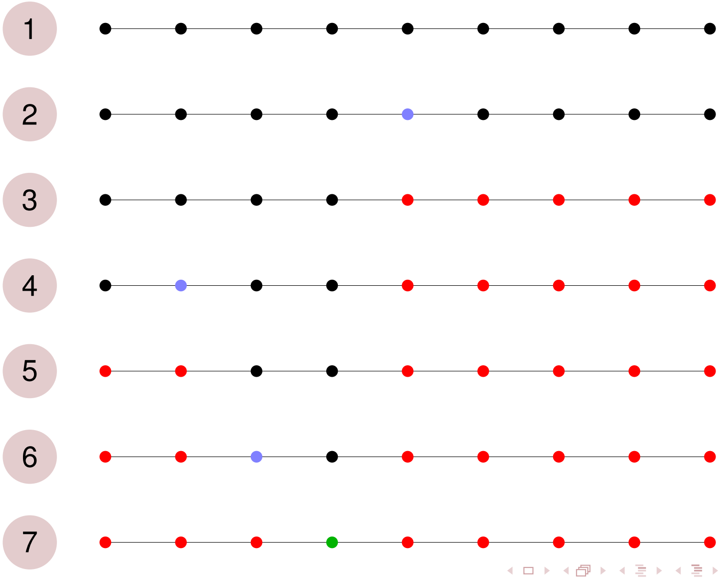
One dimensional solution — a binary search



One dimensional solution — a binary search



One dimensional solution — a binary search

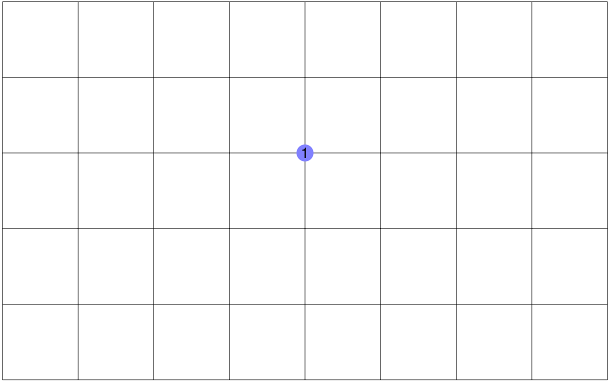


Outline

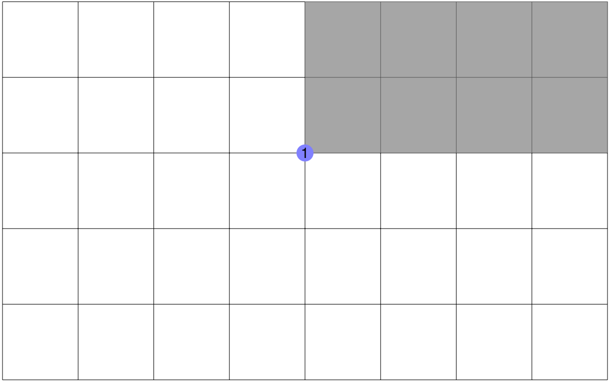
Multidimensional binary search as a game

1. A game between Algorithm and Adversary
2. A board: d-dimensional grid $n_1 \times n_2 \times \dots n_d$
3. Adversary hides a target in some point of the grid
4. At each step Algorithm picks a point $(x_1, x_2, \dots x_d)$
5. Adversary answers with (d-dimensional) interval with one end at $(x_1, x_2, \dots x_d)$ and another at some corner, where there is no target.
6. The Algorithm' goal is to find the target as soon as possible
7. Adversary wants to play long
8. It is good to think that there is no target point, but the area of possible target places shrinks at each step
9. Let $T(n_1, n_2 \dots n_d)$ be the number of steps provided both player play optimally. Surely $T(n) = \Theta(\log n)$.

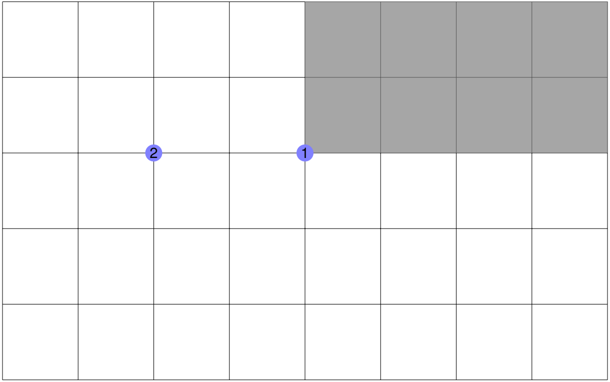
2D example



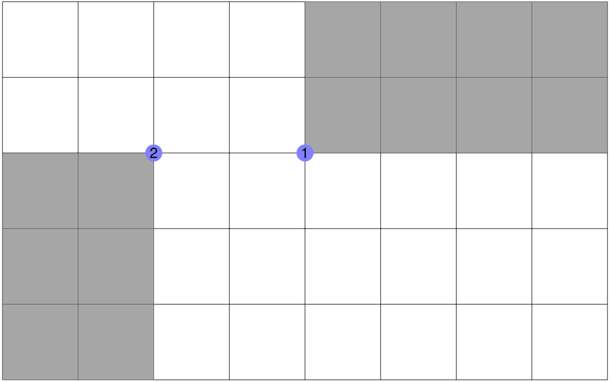
2D example



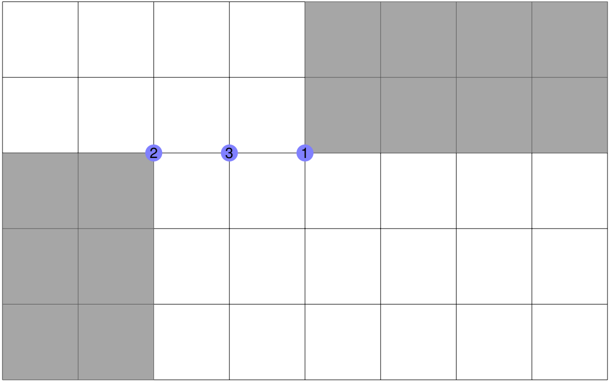
2D example



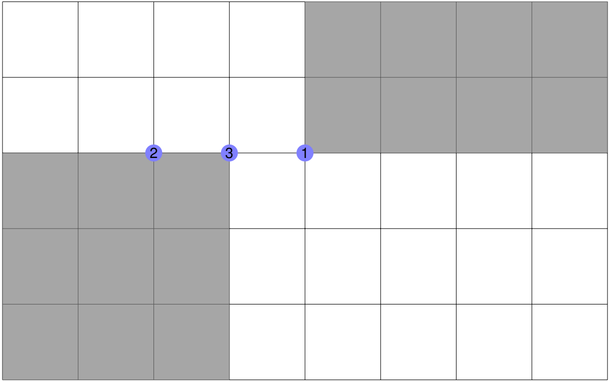
2D example



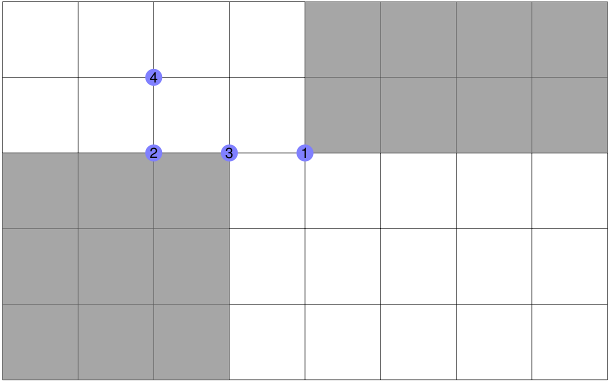
2D example



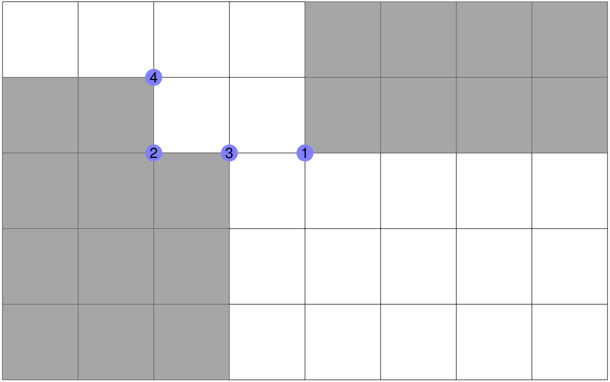
2D example



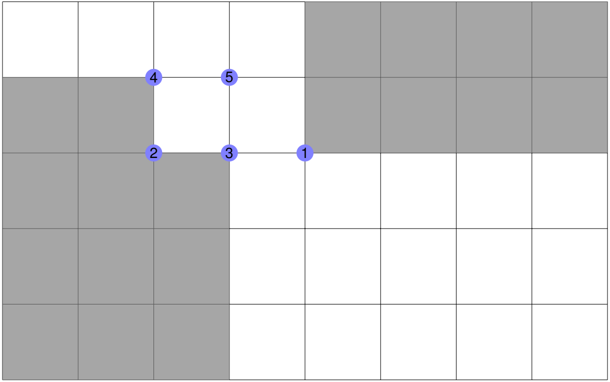
2D example



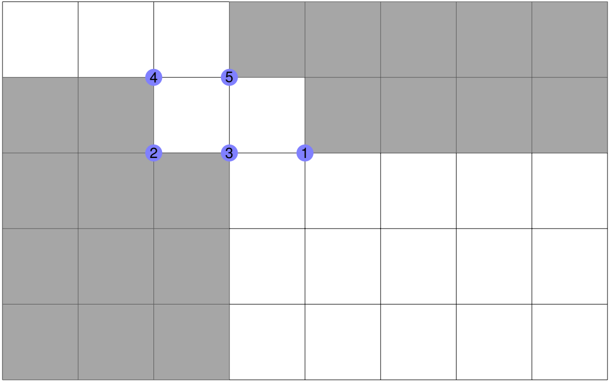
2D example



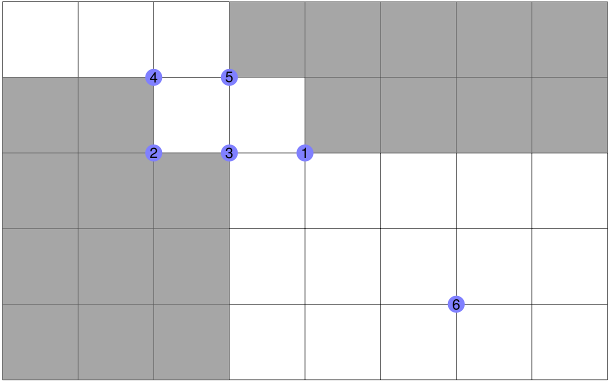
2D example



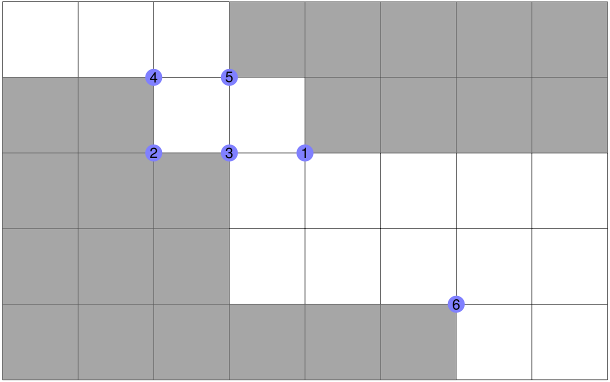
2D example



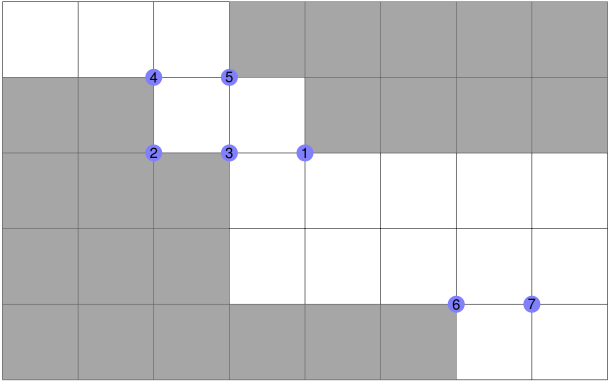
2D example



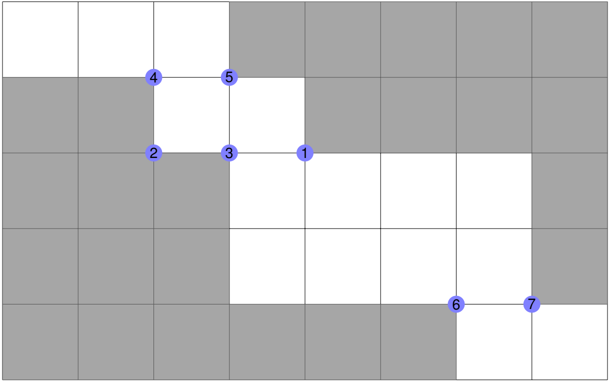
2D example



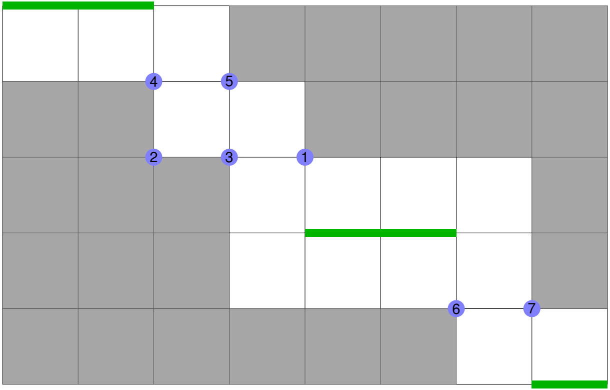
2D example



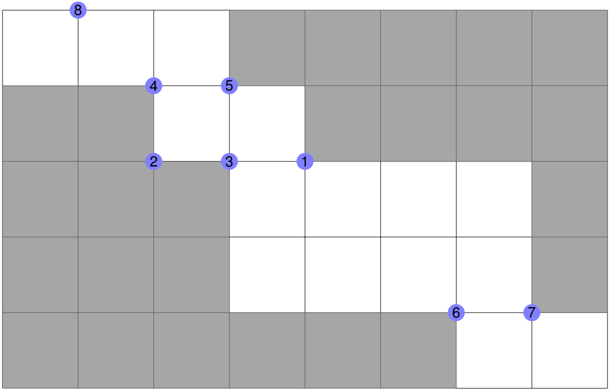
2D example



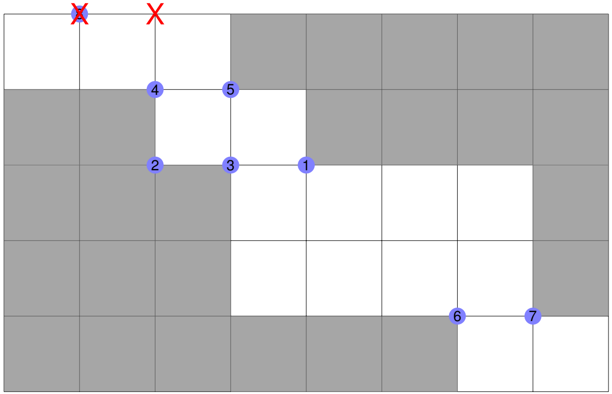
2D example



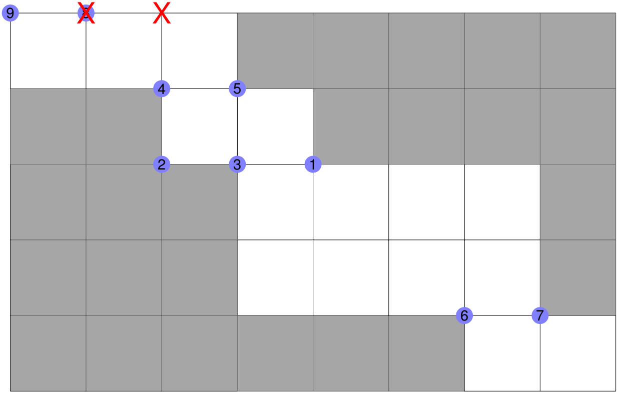
2D example



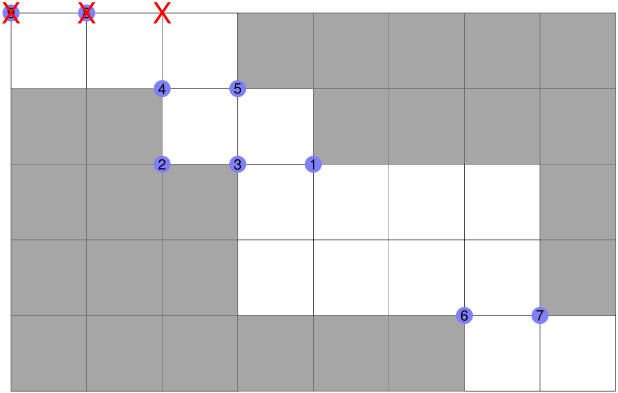
2D example



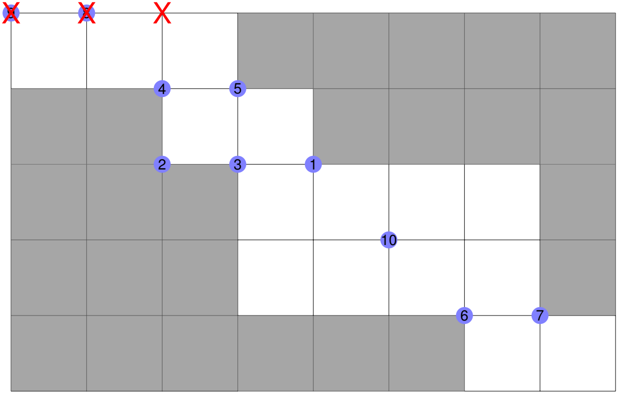
2D example



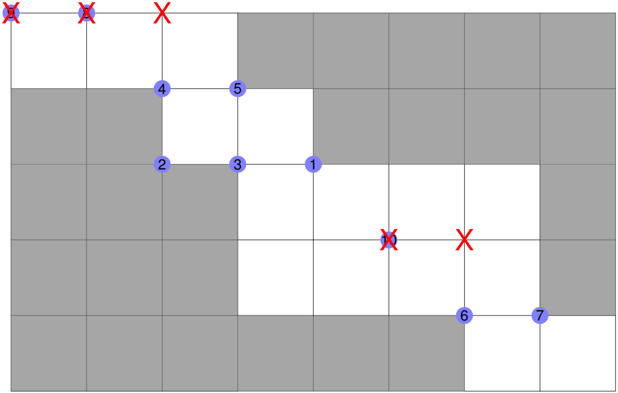
2D example



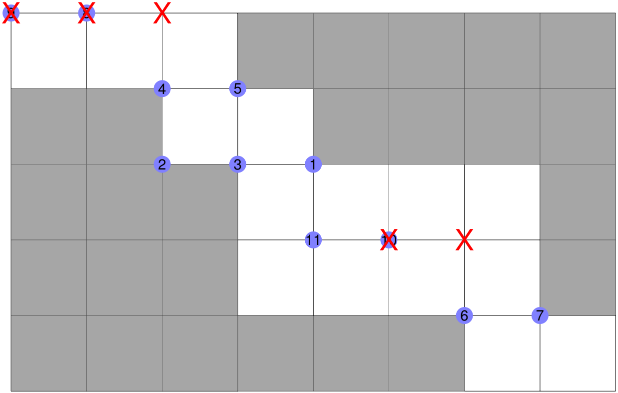
2D example



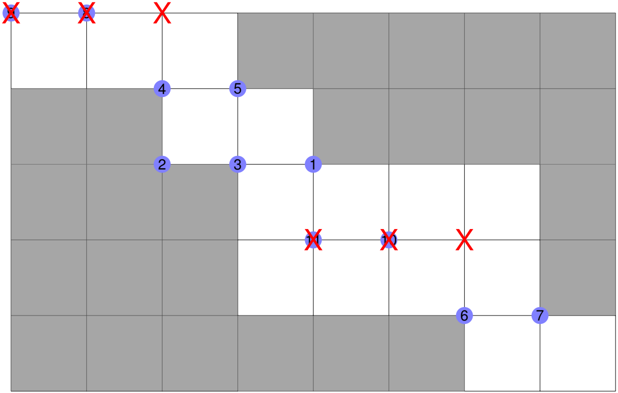
2D example



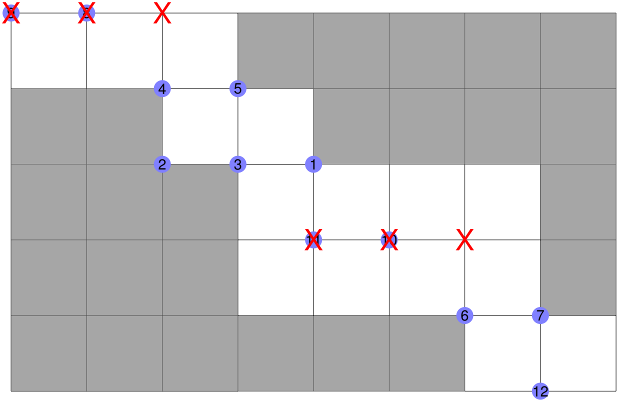
2D example



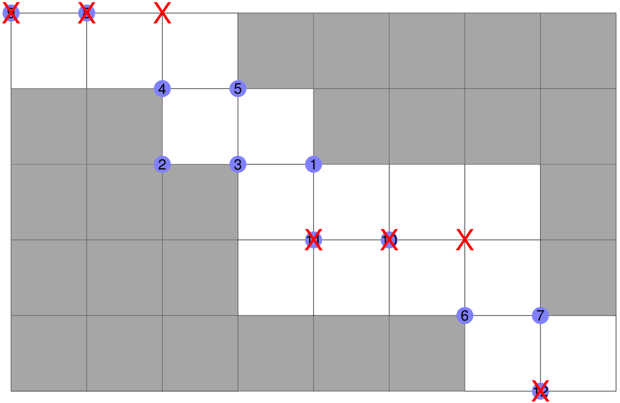
2D example



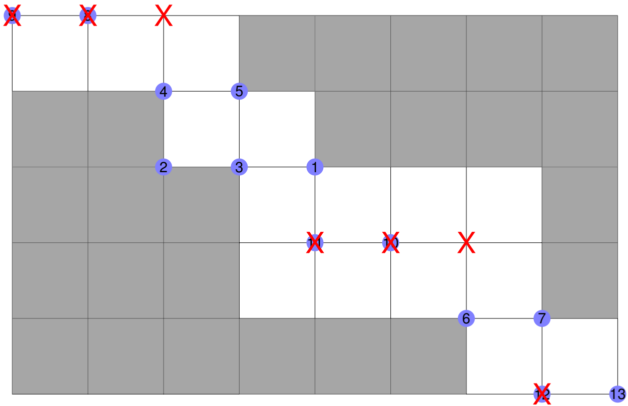
2D example



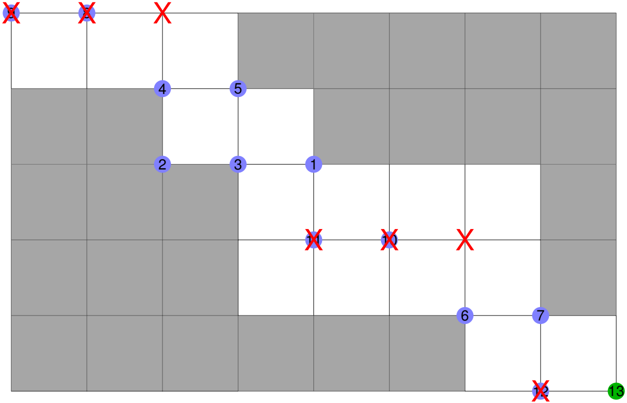
2D example



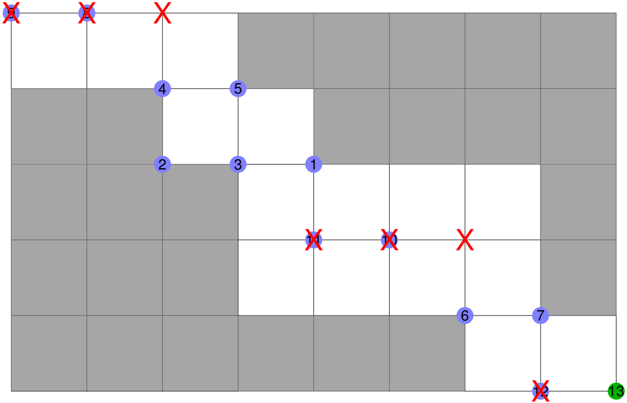
2D example



2D example



2D example

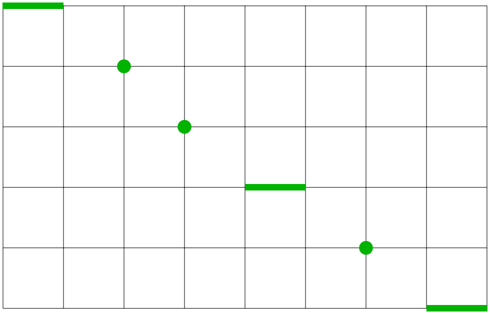


Theorem (Dereniowski, G., Wróbel)

Given $n \leq m$, $T(n,m) = \Theta(n \log(\frac{m}{n} + 1))$.

Informal proof (2D) $T(n,m) = \Theta(n \log(\frac{m}{n} + 1))$

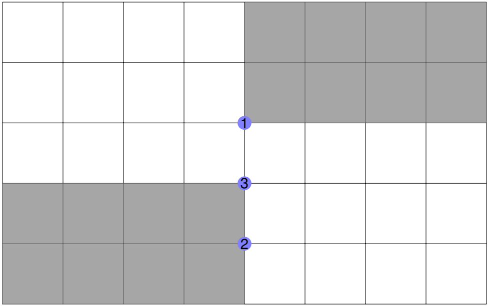
From below $T(n,m) = \Omega(n \log(\frac{m}{n} + 1))$:



- n (independent) segments with roughly $\frac{m}{n}$ elements,
- for each perform binary search with $\log_2 \frac{m}{n} + 1$ queries,

Informal proof (2D) $T(n, m) = \Theta(n \log(\frac{m}{n} + 1))$

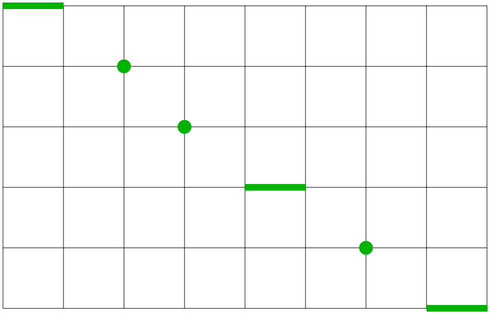
From above $T(n, m) = \mathcal{O}(n \log(\frac{m}{n} + 1))$:



- $\log n$ queries to split the board into 2 (independent) pieces,
- for Adversary it is optimal to have them of the same size,
- hence $T(n, m) \leq \log n + 2T(\frac{n}{2}, \frac{m}{2})$
- after $\mathcal{O}(n)$ steps we are left with diagonal from the lower bound

Informal proof (2D) $T(n,m) = \Theta(n\log(\frac{m}{n}+1))$

From above $T(n,m) = \mathcal{O}(n\log(\frac{m}{n}+1))$:



- $\log n$ queries to split the board into 2 (independent) pieces,
- for Adversary optimal is to have them of the same size,
- hence $T(n,m) \leq \log n + 2T(\frac{n}{2},\frac{m}{2})$
- after $\mathcal{O}(n)$ steps we are left with diagonal from the lower bound

d-dimensional case

Theorem (Dereniowski, G., Wróbel)

Given d and $n_1, \ldots n_d$, such that $n_i \leq n_{i+1}$ there is

$$T(n_1, \ldots, n_d) = \mathcal{O}\left(\left(\prod_{i=1}^{d-1} n_i\right) \log\left(\frac{n_d}{n_{d-1}} + 1\right)\right).$$

For proof: Algorithm consider n_1 $(d-1)$-dimensional cubes.

Theorem (Dereniowski, G., Wróbel)

For fixed d there is $T(n, \ldots, n) \geq \frac{2}{d-1} n^{d-1}$,
i.e $T(n, \ldots, n) = \Omega\left(\frac{n^{d-1}}{d}\right)$.

For proof: Adversary hides target at the hyperplane given by

$$x_d = n - \left\lceil \frac{x_1 + \cdots + x_{d-1}}{d-1} \right\rceil.$$

d-dimensional case

Theorem (Dereniowski, G., Wróbel)

Given d and $n_1, \ldots n_d$, such that $n_i \leq n_{i+1}$ there is

$$T(n_1, \ldots, n_d) = \mathcal{O}\left(\left(\prod_{i=1}^{d-1} n_i\right) \log\left(\frac{n_d}{n_{d-1}} + 1\right)\right).$$

For proof: Algorithm consider n_1 $(d-1)$-dimensional cubes.

Theorem (Dereniowski, G., Wróbel)

For fixed d there is $T(n, \ldots, n) \geq \frac{2}{d-1} n^{d-1}$,
i.e $T(n, \ldots, n) = \Omega\left(\frac{n^{d-1}}{d}\right)$.

For proof: Adversary hides target at the hyperplane given by

$$x_d = n - \left\lceil \frac{x_1 + \cdots + x_{d-1}}{d-1} \right\rceil.$$

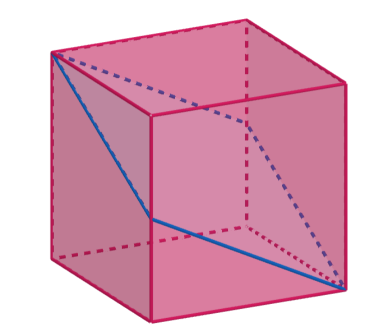
d-dimensional case

Theorem (Dereniowski, G., Wróbel)

Given d and … *there is*

$$T(n_1, \ldots \frac{n_d}{n_{d-1}} + 1 \Big) \Big) .$$

For proof: Alg … imensional cubes.



Theorem (De …

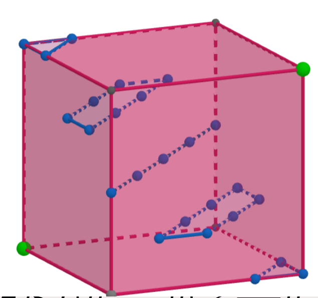
For fixed d th … ,

i.e $T(n, \ldots, n) = \Omega\left(\frac{n}{d}\right)$.

For proof: Adversary hides target at the hyperplane given by

$$x_d = n - \left\lceil \frac{x_1 + \cdots + x_{d-1}}{d-1} \right\rceil .$$

d-dimensional case



Theorem (Dereniowski, G., Wróbel)

Given d and n_1 … $_{+1}$ there is

$$T(n_1, \ldots \quad \ldots \left(\frac{n_d}{n_{d-1}} + 1\right)\Bigg) .$$

For proof: Algo… -dimensional cubes.

Theorem (Dere…

For fixed d there is $T(n, \ldots, n) \geq \frac{\ldots}{d-1} n^{\ldots -1}$,

i.e $T(n, \ldots, n) = \Omega\left(\frac{n^{d-1}}{d}\right)$.

For proof: Adversary hides target at the hyperplane given by

$$x_d = n - \left\lceil \frac{x_1 + \cdots + x_{d-1}}{d-1} \right\rceil .$$

d-dimensional case — further work

Theorem (Dereniowski, G., Wróbel)

For fixed d there is $T(n,\ldots,n) = \Omega\left(\frac{n^{d-1}}{d}\right)$ and $T(n,\ldots,n) = \mathcal{O}\left(n^{d-1}\right)$.

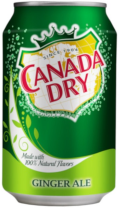
To do:

- Fill the gap.
- Lower bound when $(n_1,\ldots,n_d) \neq (n,\ldots,n)$ (we did it for $d \leq 3$).

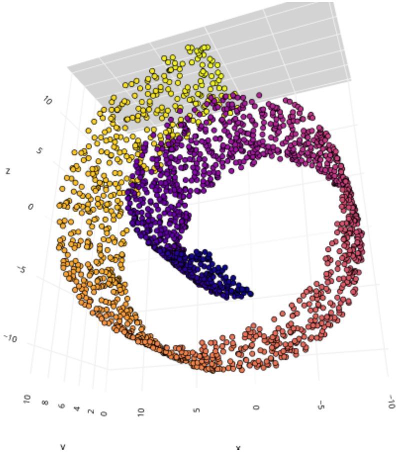
Outline

Why graphs? Graphical embeddings of data

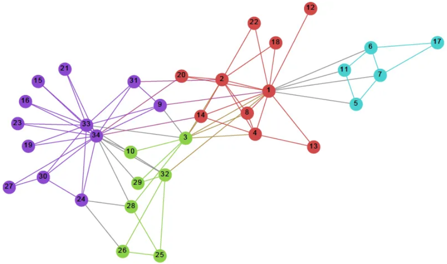
Why graphs? Graphical embeddings of data



Why graphs? Graphical embeddings of data



Why graphs? Graphical embeddings of data



Edge and pair queries (Dereniowski, G., Prałat [2023])

1. A game between Algorithm and Adversary
2. A board: graph G
3. Adversary hides a target at some vertex $x \in V(G)$
4. At each step Algorithm picks two (adjacent) vertices u, v
5. Adversary answers which vertex is closer to x breaking ties arbitrary
6. The Algorithm' goal is to find the target as soon as possible
7. Adversary wants to play long
8. It is good to think that there is no target vertex, but the area of possible target places shrinks at each step
9. Let pq(G) (eq(G)) be the number of rounds provided both player play optimally. Surely pq(G) $\leq$ eq(G).

Edge and pair queries — simple properties

Observation

For any connected graph G on n vertices,

$$\log_2 n \leq \mathrm{pq}(G) \leq \mathrm{eq}(G) \leq n - 1.$$

In fact, there exists a strategy of the algorithm that in each round eliminates at least one vertex from the search space.

Lower bound is achieved by P_n (binary search), upper — by K_n and $K_{1,n-1}$.

Edge and pair queries — simple properties

Observation

For any connected graph G on n vertices,

$$\log_2 n \le \mathrm{pq}(G) \le \mathrm{eq}(G) \le n-1.$$

In fact, there exists a strategy of the algorithm that in each round eliminates at least one vertex from the search space.

Lower bound is achieved by P_n (binary search), upper — by K_n and $K_{1,n-1}$.

Edge and pair queries — simple properties

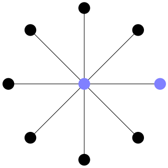
Observation

For any connected graph G on n vertices,

$$\log_2 n \leq \mathrm{pq}(G) \leq \mathrm{eq}(G) \leq n - 1.$$

In fact, there exists a strategy of the algorithm that in each round eliminates at least one vertex from the search space.

Lower bound is achieved by P_n (binary search), upper — by K_n and $K_{1,n-1}$.



Edge and pair queries — simple properties

Observation

For any connected graph G on n vertices,

$$\log_2 n \leq \mathrm{pq}(G) \leq \mathrm{eq}(G) \leq n - 1.$$

In fact, there exists a strategy of the algorithm that in each round eliminates at least one vertex from the search space.

Lower bound is achieved by P_n (binary search), upper — by K_n and $K_{1,n-1}$.

Observation

For any $n \in \mathbb{N} \setminus \{1\}$, there exists a graph G on $\Theta(n)$ vertices such that $\mathrm{eq}(G) = \Omega(n)$ and $\mathrm{pq}(G) = \mathcal{O}(\log n)$.

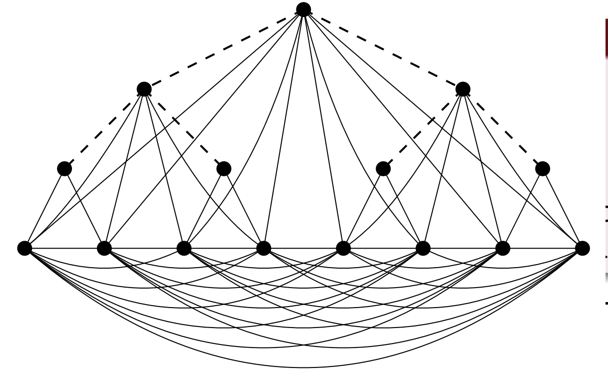
Edge and pair queries — simple properties

Observat

For any c

In fact, th ... ch
round eli ... ce.

Lower bo ... - by K_n
and $K_{1,n-}$



Observation

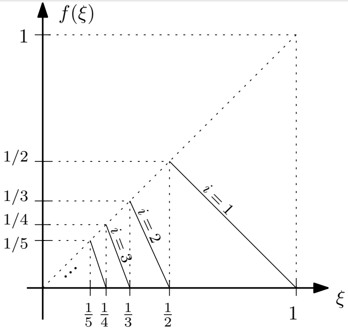
For any $n \in \mathbb{N} \setminus \{1\}$, there exists a graph G on $\Theta(n)$ vertices such that $\mathrm{eq}(G) = \Omega(n)$ and $\mathrm{pq}(G) = \mathcal{O}(\log n)$.

Edge and pair queries — results

Theorem (Dereniowski, G., Prałat [2023])

Suppose that $pn = n^{\xi+o(1)}$, where $\xi \in (\frac{1}{i+1}, \frac{1}{i})$ for some $i \in \mathbb{N}$. Then, a.a.s. for $G \in \mathcal{G}(n,p)$.

$$\mathrm{pq}(G) = \Theta\Big(\mathrm{eq}(G)\Big) = \Theta\left(n^{f(\xi)+o(1)}\right), \textit{ for } f$$



Edge and pair queries — results

Theorem (Dereniowski, G., Prałat [2023])

Suppose that $pn = n^{\xi+o(1)}$, where $\xi \in (\frac{1}{i+1}, \frac{1}{i})$ for some $i \in \mathbb{N}$. Then, a.a.s. for $G \in \mathcal{G}(n,p)$.

$$\mathrm{pq}(G) = \Theta\Big(\mathrm{eq}(G)\Big) = \Theta\left(n^{f(\xi)+o(1)}\right), \textit{ for } f$$

Theorem (Dereniowski, G., Prałat [2023])

For fixed $p \in (0,1)$ a.a.s. for $G \in \mathcal{G}(n,p)$.

$$\mathrm{pq}(G) = \Theta\Big(\mathrm{eq}(G)\Big) = \Theta\left(\log n\right).$$

Theorem (Dereniowski, G., Prałat [2023])

Calculating $\mathrm{pq}(G)$ or $\mathrm{eq}(G)$ is $\mathcal{NP}$-hard, even for graphs of diameter at most 3.

Edge and pair queries — questions

Question

What happen for particular graph classes?
(eg. grids, intersection graphs)

Edge and pair queries — questions

Question

What happen for particular graph classes?
(eg. grids, intersection graphs)

Thank you for the attention!

Source of pictures:
www.homegrounds.co (Coffee Compass)
some commercials (cold drink samples)
towardsdatascience.com (graph embeddings)