The Rectilinear Marco Polo Problem

Ofek Gila[1], Michael T. Goodrich[1], Zahra Hadizadeh[2], Daniel S. Hirschberg[1], and Shayan Taherijam[1]

[1]University of California, Irvine

[2]University of Rochester

CCCG, 2025

The Marco Polo Problem I

- Point of Interest (POI) ✗
- ✗ within distance n from origin

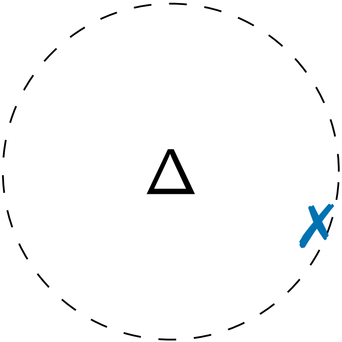


Figure 1: A search area in L_2.

The Marco Polo Problem I

- Point of Interest (POI) ✗
- ✗ within distance n from origin
- Probes with radius d, $p(x, y, d)$

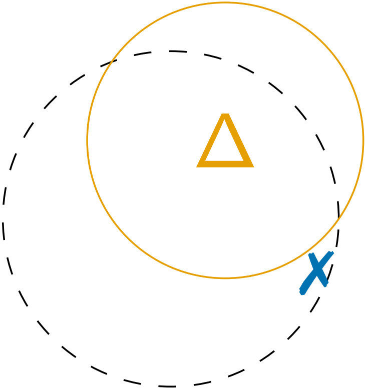


Figure 1: A search area in L_2.

The Marco Polo Problem I

- Point of Interest (POI) ✗
- ✗ within distance n from origin
- Probes with radius d, $p(x, y, d)$
- Probe until 'finding' ✗

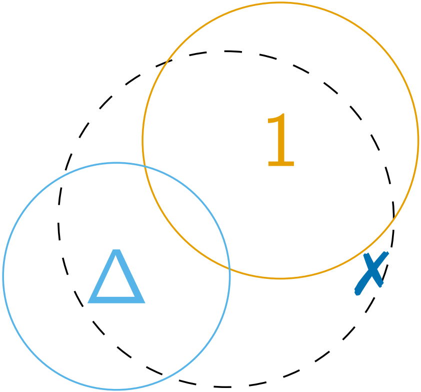


Figure 1: A search area in L_2.

The Marco Polo Problem I

- Point of Interest (POI) ✗
- ✗ within distance n from origin
- Probes with radius d, $p(x, y, d)$
- Probe until 'finding' ✗.

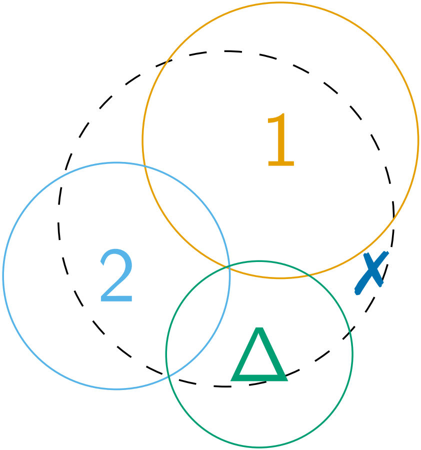


Figure 1: A search area in L_2.

The Marco Polo Problem I

- Point of Interest (POI) ✗
- ✗ within distance n from origin
- Probes with radius d, $p(x, y, d)$
- Probe until 'finding' ✗..

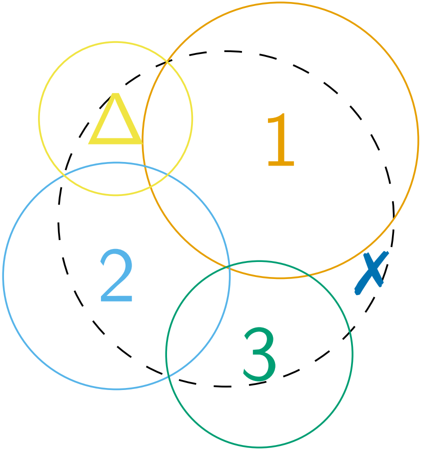


Figure 1: A search area in L_2.

The Marco Polo Problem I

- Point of Interest (POI) ✗
- ✗ within distance n from origin
- Probes with radius d, $p(x, y, d)$
- Probe until 'finding' ✗... ✓
- 'finding': distance $\Delta \leftrightarrow$ ✗ ≤ 1

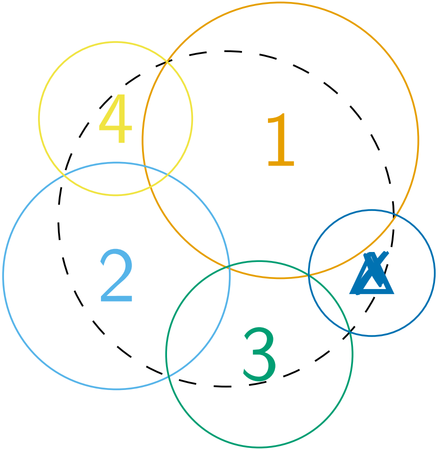


Figure 1: A search area in L_2.

The Marco Polo Problem I

- Point of Interest (POI) ✗
- ✗ within distance n from origin
- Probes with radius d, $p(x, y, d)$
- Probe until 'finding' ✗... ✓
- 'finding': distance $\Delta \leftrightarrow$ ✗ ≤ 1
- Δ must know this!

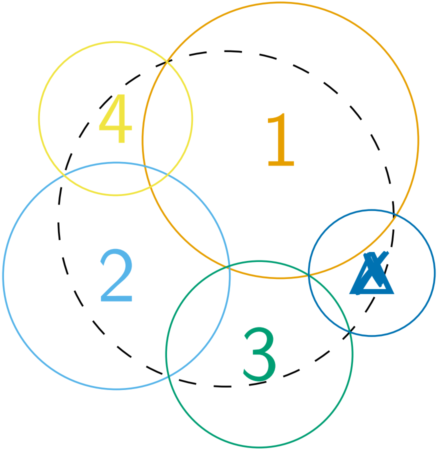


Figure 1: A search area in L_2.

The Marco Polo Problem I

- Point of Interest (POI) ✗
- ✗ within distance n from origin
- Probes with radius d, $p(x, y, d)$
- Probe until 'finding' ✗... ✓
- 'finding': distance $\Delta \leftrightarrow$ ✗ ≤ 1
- Δ must know this!
- Variants:
  - $\#$ of POIs present (k)

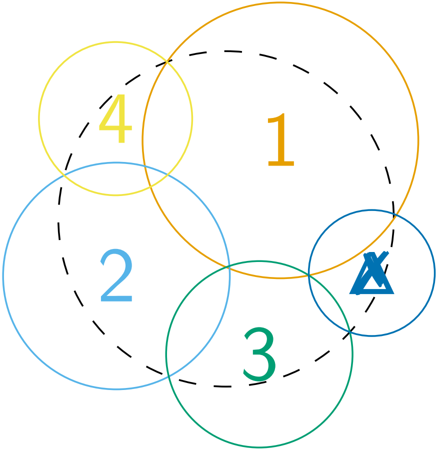


Figure 1: A search area in L_2.

The Marco Polo Problem I

- Point of Interest (POI) ✗
- ✗ within distance n from origin
- Probes with radius d, $p(x, y, d)$
- Probe until 'finding' ✗... ✓
- 'finding': distance $\Delta \leftrightarrow$ ✗ ≤ 1
- Δ must know this!
- Variants:
  - $\#$ of POIs present (k)
  - find all POIs

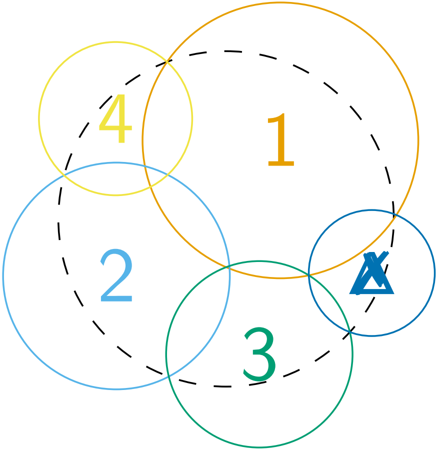


Figure 1: A search area in L_2.

The Marco Polo Problem I

- Point of Interest (POI) ✗
- ✗ within distance n from origin
- Probes with radius d, $p(x, y, d)$
- Probe until 'finding' ✗... ✓
- 'finding': distance $\Delta \leftrightarrow$ ✗ ≤ 1
- Δ must know this!
- Variants:
  - $\#$ of POIs present (k)
  - find all POIs
  - Distance metrics (L_1, L_∞)

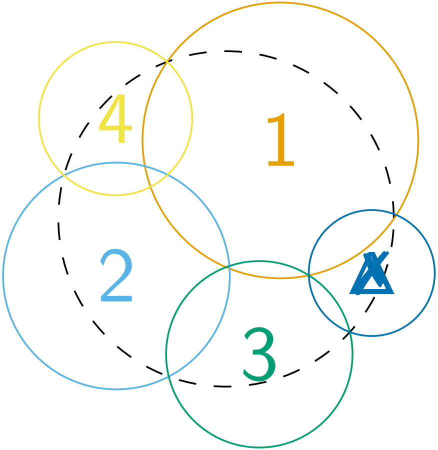


Figure 1: A search area in L_2.

The Marco Polo Problem I

- Point of Interest (POI) ✗
- ✗ within distance n from origin
- Probes with radius d, $p(x, y, d)$
- Probe until 'finding' ✗... ✓
- 'finding': distance $\Delta \leftrightarrow$ ✗ ≤ 1
- Δ must know this!
- Variants:
    - # of POIs present (k)
    - find all POIs
    - Distance metrics (L_1, L_∞)
    - # of dimensions (2D, 3D, ...?)

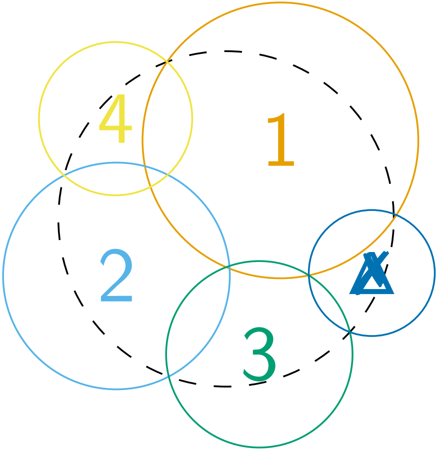


Figure 1: A search area in L_2.

The Marco Polo Problem I

- Point of Interest (POI) ✗
- ✗ within distance n from origin
- Probes with radius d, $p(x, y, d)$
- Probe until 'finding' ✗... ✓
- 'finding': distance $\Delta \leftrightarrow$ ✗ ≤ 1
- Δ must know this!
- Variants:
  - # of POIs present (k)
  - find all POIs
  - Distance metrics (L_1, L_∞)
  - # of dimensions (2D, 3D, ...?)
  - Probe response (T/F, d, i)

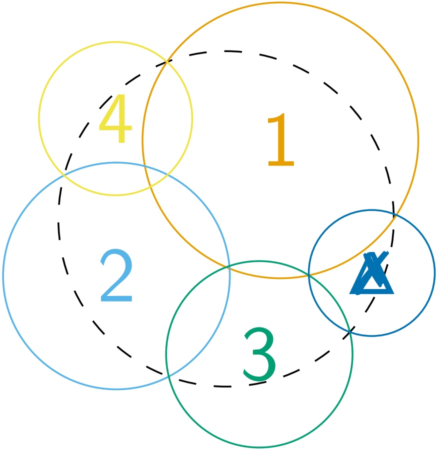


Figure 1: A search area in L_2.

The Marco Polo Problem I

- Point of Interest (POI) ✗
- ✗ within distance n from origin
- Probes with radius d, $p(x, y, d)$
- Probe until 'finding' ✗... ✓
- 'finding': distance $\Delta \leftrightarrow$ ✗ ≤ 1
- Δ must know this!
- Variants:
  - # of POIs present (k)
  - find all POIs
  - Distance metrics (L_1, L_∞)
  - # of dimensions (2D, 3D, ...?)
  - Probe response (T/F, d, i)
  - Δ's memory if any

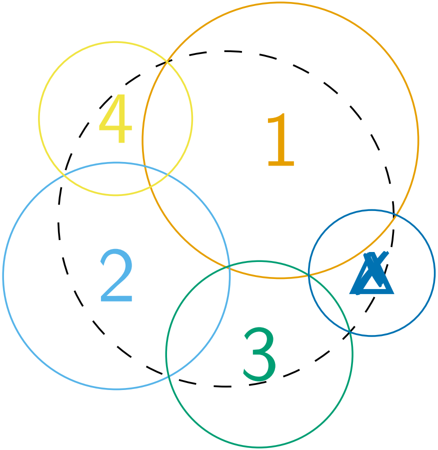


Figure 1: A search area in L_2.

The Marco Polo Problem I

- Point of Interest (POI) ✗
- ✗ within distance n from origin
- Probes with radius d, $p(x, y, d)$
- Probe until 'finding' ✗... ✓
- 'finding': distance $\Delta \leftrightarrow$ ✗ ≤ 1
- Δ must know this!
- Variants:
  - # of POIs present (k)
  - find all POIs
  - Distance metrics (L_1, L_∞)
  - # of dimensions (2D, 3D, ...?)
  - Probe response (T/F, d, i)
  - Δ's memory if any
  - multiple Δ, etc...

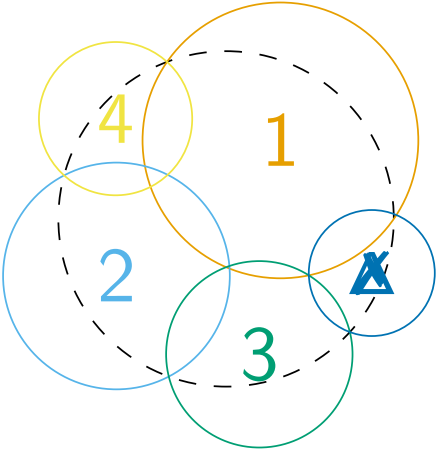


Figure 1: A search area in L_2.

The Marco Polo Problem II

- Variants:
  - ☑ # of POIs present (k)
  - ☑ find all POIs
  - ☐ Distance metrics (L_1, L_∞)
  - ☐ # of dimensions (2D, 3D, ...?)
  - ☐ Probe response (T/F, d, i)
  - ☐ Δ's memory if any
  - ☐ multiple Δ, etc...
- Effectiveness metrics:

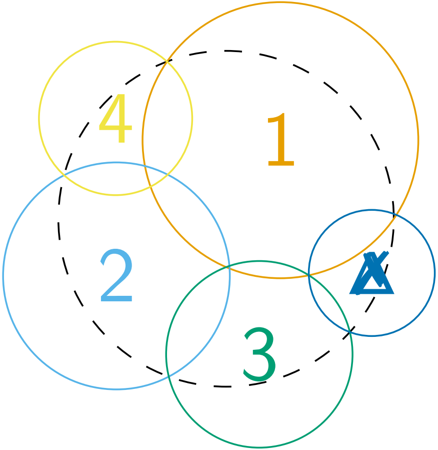


Figure 2: A search area in L_2.

The Marco Polo Problem II

- Variants:
  - ☑ # of POIs present (k)
  - ☑ find all POIs
  - ☐ Distance metrics (L_1, L_∞)
  - ☐ # of dimensions (2D, 3D, ...?)
  - ☐ Probe response (T/F, d, i)
  - ☐ Δ's memory if any
  - ☐ multiple Δ, etc...
- Effectiveness metrics:
  - ☐ # of probes, $P(n)$

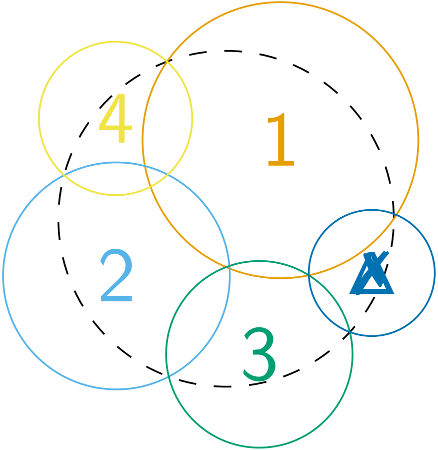


Figure 2: A search area in L_2.

The Marco Polo Problem II

- Variants:
  - ☑ # of POIs present (k)
  - ☑ find all POIs
  - ☐ Distance metrics (L_1, L_∞)
  - ☐ # of dimensions (2D, 3D, ...?)
  - ☐ Probe response (T/F, d, i)
  - ☐ Δ's memory if any
  - ☐ multiple Δ, etc...
- Effectiveness metrics:
  - ☐ # of probes, $P(n)$
  - ☐ Distance traveled by Δ, $D(n)$

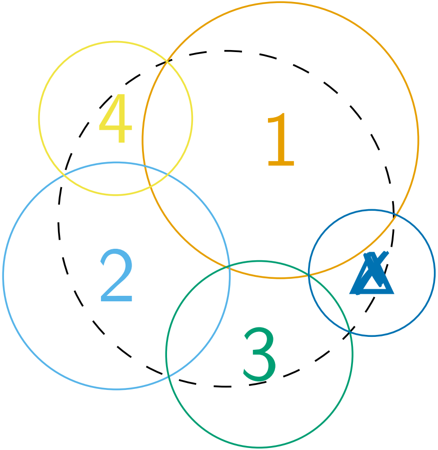


Figure 2: A search area in L_2.

The Marco Polo Problem II

- Variants:
  - ☑ # of POIs present (k)
  - ☑ find all POIs
  - ☐ Distance metrics (L_1, L_∞)
  - ☐ # of dimensions (2D, 3D, ...?)
  - ☐ Probe response (T/F, d, i)
  - ☐ Δ's memory if any
  - ☐ multiple Δ, etc...
- Effectiveness metrics:
  - ☐ # of probes, $P(n)$
  - ☐ Distance traveled by Δ, $D(n)$
  - ☑ # of POI responses, $R(n)$

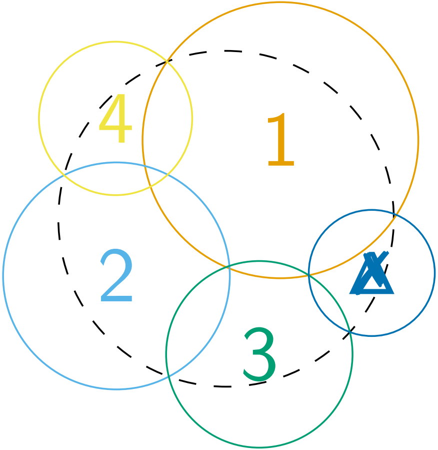


Figure 2: A search area in L_2.

The Marco Polo Problem II

- Variants:
  - ☑ # of POIs present (k)
  - ☑ find all POIs
  - ☐ Distance metrics (L_1, L_∞)
  - ☐ # of dimensions (2D, 3D, ...?)
  - ☐ Probe response (T/F, d, i)
  - ☐ Δ's memory if any
  - ☐ multiple Δ, etc...
- Effectiveness metrics:
  - ☐ # of probes, $P(n)$
  - ☐ Distance traveled by Δ, $D(n)$
  - ☑ # of POI responses, $R(n)$
  - Input sensitivity?

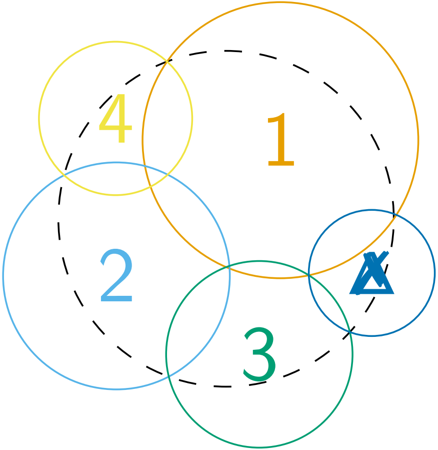


Figure 2: A search area in L_2.

The Marco Polo Problem II

- Variants:
  - ☑ # of POIs present (k)
  - ☑ find all POIs
  - ☐ Distance metrics (L_1, L_∞)
  - ☐ # of dimensions (2D, 3D, ...?)
  - ☐ Probe response (T/F, d, i)
  - ☐ Δ's memory if any
  - ☐ multiple Δ, etc...
- Effectiveness metrics:
  - ☐ # of probes, $P(n)$
  - ☐ Distance traveled by Δ, $D(n)$
  - ☑ # of POI responses, $R(n)$
  - Input sensitivity?
    - ☑ TSP tour length, `OPT`

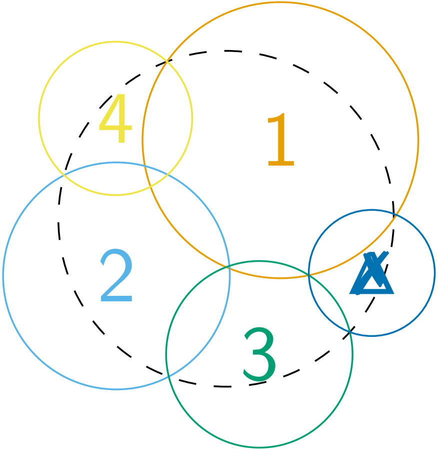


Figure 2: A search area in L_2.

The Marco Polo Problem II

- Variants:
  - [x] # of POIs present (k)
  - [x] find all POIs
  - [ ] Distance metrics (L_1, L_∞)
  - [ ] # of dimensions (2D, 3D, ...?)
  - [ ] Probe response (T/F, d, i)
  - [ ] Δ's memory if any
  - [ ] multiple Δ, etc...
- Effectiveness metrics:
  - [ ] # of probes, $P(n)$
  - [ ] Distance traveled by Δ, $D(n)$
  - [x] # of POI responses, $R(n)$
  - Input sensitivity?
    - [x] TSP tour length, `OPT`
    - [ ] Dist. to nearest POI, $\delta_{\min}$

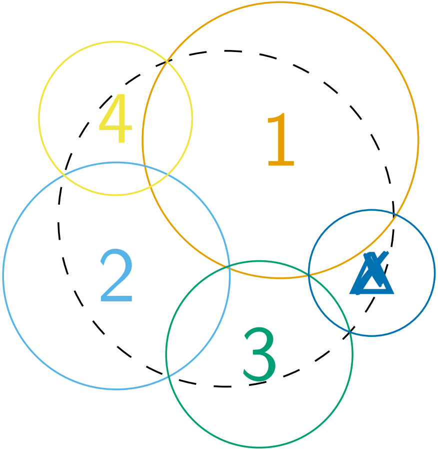


Figure 2: A search area in L_2.

The Marco Polo Problem II

- Variants:
  - ☑ # of POIs present (k)
  - ☑ find all POIs
  - ☐ Distance metrics (L_1, L_∞)
  - ☐ # of dimensions (2D, 3D, ...?)
  - ☐ Probe response (T/F, d, i)
  - ☐ Δ's memory if any
  - ☐ multiple Δ, etc...
- Effectiveness metrics:
  - ☐ # of probes, $P(n)$
  - ☐ Distance traveled by Δ, $D(n)$
  - ☑ # of POI responses, $R(n)$
  - Input sensitivity?
    - ☑ TSP tour length, `OPT`
    - ☐ Dist. to nearest POI, $\delta_{\min}$
  - ☐ Simplicity / practicality

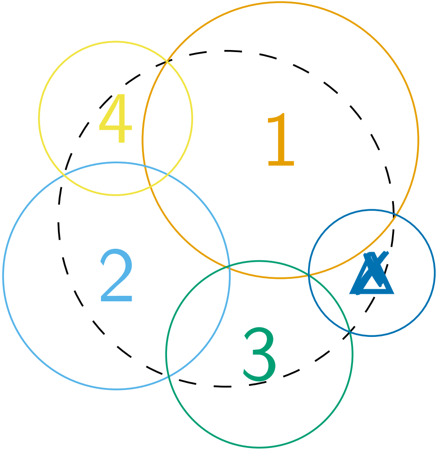


Figure 2: A search area in L_2.

The Marco Polo Problem II

- Variants:
  - ☑ # of POIs present (k)
  - ☑ find all POIs
  - ☑ Distance metrics (L_1, L_∞)
  - ☑ # of dimensions (2D, 3D, ...?)
  - ? Probe response (T/F, d, i)
  - ? Δ's memory if any
  - ? multiple Δ, etc...
- Effectiveness metrics:
  - ☑ # of probes, $P(n)$
  - ☑ Distance traveled by Δ, $D(n)$
  - ☑ # of POI responses, $R(n)$
  - Input sensitivity?
    - ☑ TSP tour length, `OPT`
    - ☑ Dist. to nearest POI, $\delta_{\min}$
  - ? Simplicity / practicality

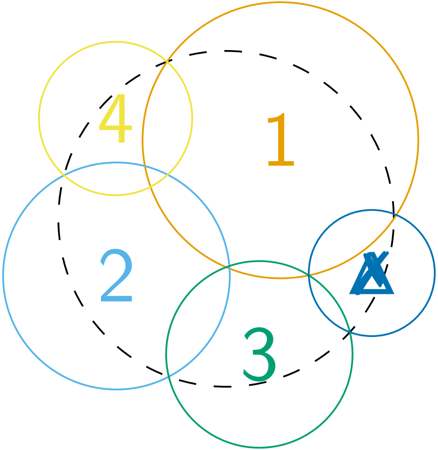
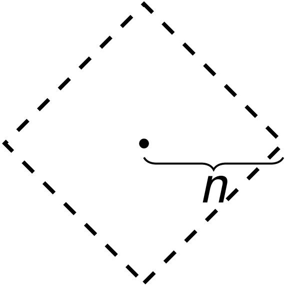


Figure 2: A search area in L_2.

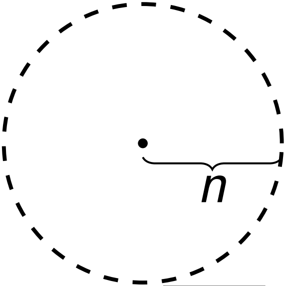
Rectilinear Distances

- $1D$: All identical



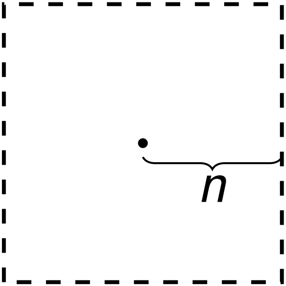
$\|x\|_1 = \sum_{i=1}^{n} |x_i|$

Figure 3: L_1 search area.



$\|x\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}$

Figure 4: L_2 search area.

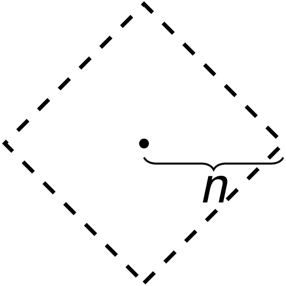


$\|x\|_\infty = \max_i |x_i|$

Figure 5: L_∞ search area.

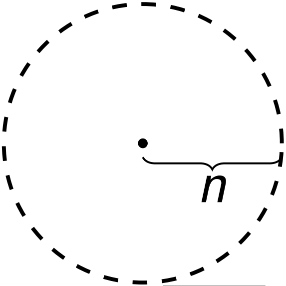
Rectilinear Distances

- $1D$: All identical
- $2D$: L_1 and L_∞ geometrically similar



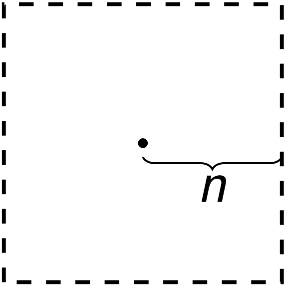
$\|x\|_1 = \sum_{i=1}^{n} |x_i|$

Figure 3: L_1 search area.



$\|x\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}$

Figure 4: L_2 search area.

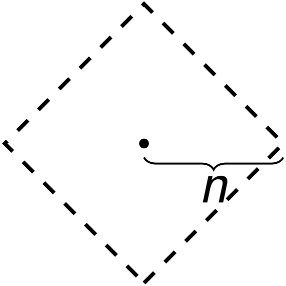


$\|x\|_\infty = \max_i |x_i|$

Figure 5: L_∞ search area.

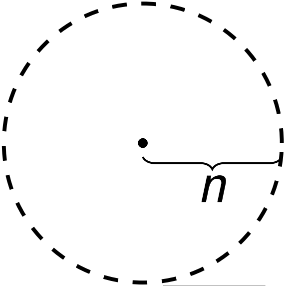
Rectilinear Distances

- $1D$: All identical
- $2D$: L_1 and L_∞ geometrically similar
- $\geq 3D$: All different



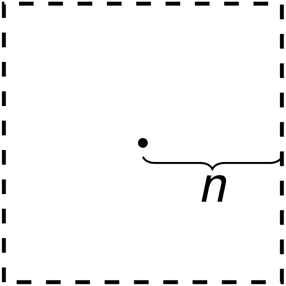
$\|x\|_1 = \sum_{i=1}^{n} |x_i|$

Figure 3: L_1 search area.



$\|x\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}$

Figure 4: L_2 search area.



$\|x\|_\infty = \max_i |x_i|$

Figure 5: L_∞ search area.

Rectilinear Strategies: Quadrant Algorithm

- Consider a 2×2 lattice

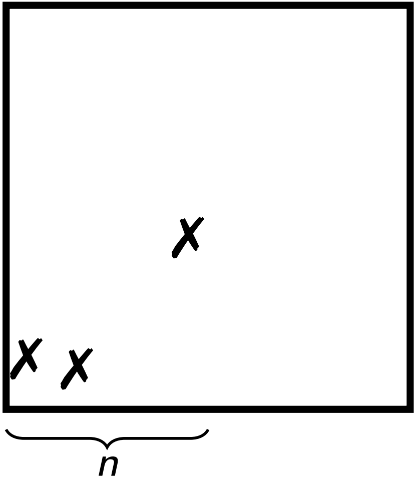


Figure 6: Quadrant Search

Rectilinear Strategies: Quadrant Algorithm

- Consider a 2×2 lattice
- Probe each quadrant.

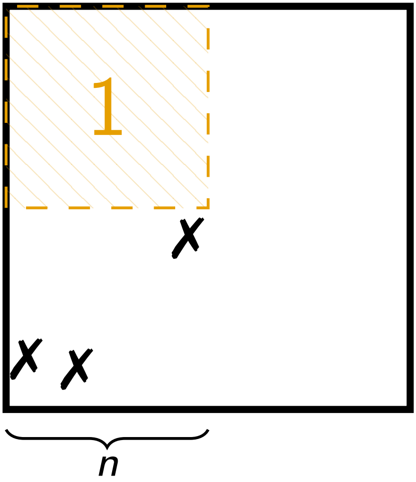


Figure 6: Quadrant Search

Rectilinear Strategies: Quadrant Algorithm

- Consider a 2×2 lattice
- Probe each quadrant..

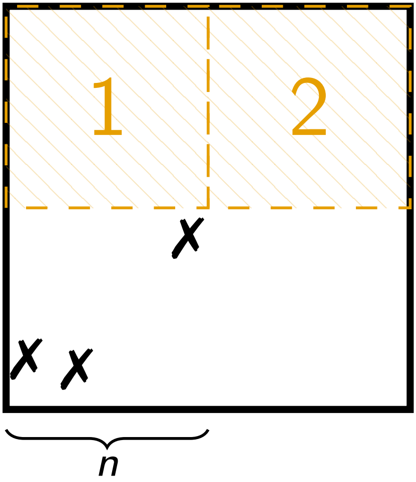


Figure 6: Quadrant Search

Rectilinear Strategies: Quadrant Algorithm

- Consider a 2×2 lattice
- Probe each quadrant...
- If 3 fail, POI/s *must* be in final

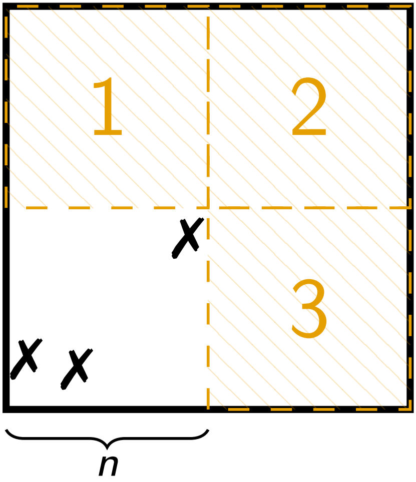


Figure 6: Quadrant Search

Rectilinear Strategies: Quadrant Algorithm

- Consider a 2×2 lattice
- Probe each quadrant...
- If 3 fail, POI/s *must* be in final
- Continue recursively!

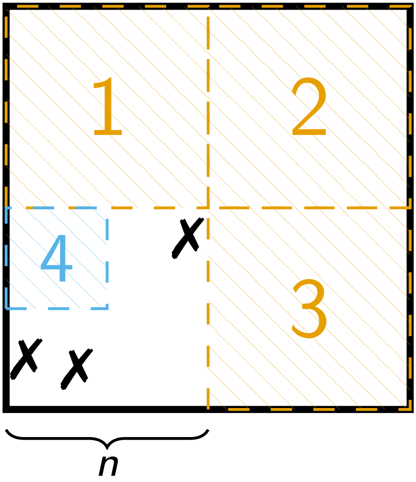


Figure 6: Quadrant Search

Rectilinear Strategies: Quadrant Algorithm

- Consider a 2×2 lattice
- Probe each quadrant...
- If 3 fail, POI/s *must* be in final
- Continue recursively!

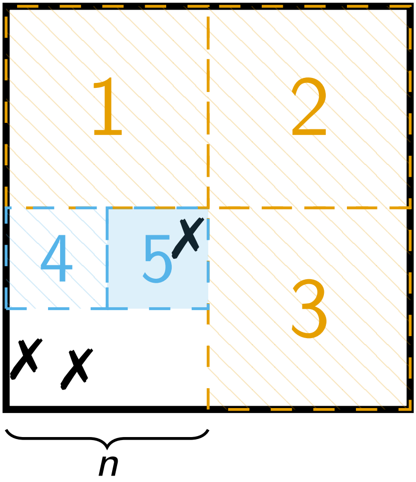


Figure 6: Quadrant Search

Rectilinear Strategies: Quadrant Algorithm

- Consider a 2×2 lattice
- Probe each quadrant...
- If 3 fail, POI/s *must* be in final
- Continue recursively!

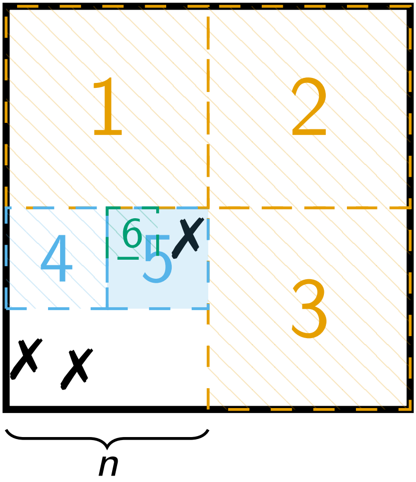


Figure 6: Quadrant Search

Rectilinear Strategies: Quadrant Algorithm

- Consider a 2×2 lattice
- Probe each quadrant...
- If 3 fail, POI/s *must* be in final
- Continue recursively!

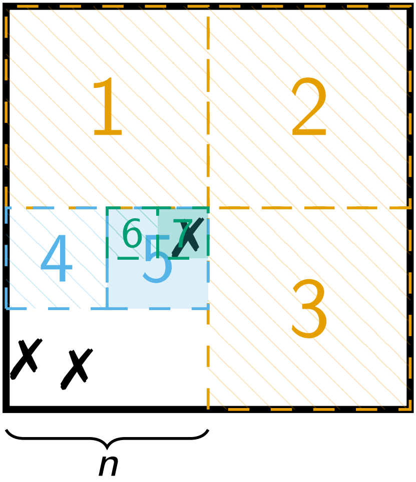


Figure 6: Quadrant Search

Rectilinear Strategies: Quadrant Algorithm

- Consider a 2×2 lattice
- Probe each quadrant…
- If 3 fail, POI/s *must* be in final
- Continue recursively!
- After 3 probes… distance halved!

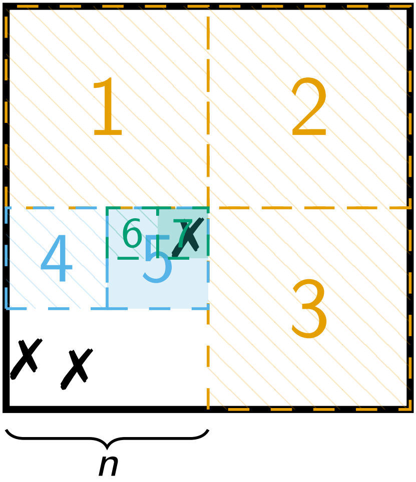


Figure 6: Quadrant Search

Rectilinear Strategies: Quadrant Algorithm

- Consider a 2×2 lattice
- Probe each quadrant...
- If 3 fail, POI/s *must* be in final
- Continue recursively!
- After 3 probes... distance halved!
- $P(n) \leq {\color{red}3}\lceil \log n \rceil$

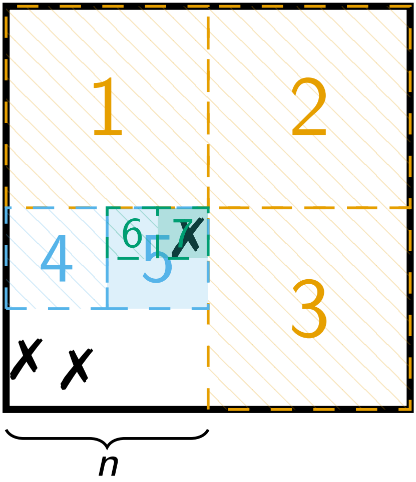


Figure 6: Quadrant Search

Rectilinear Strategies: Quadrant Algorithm

- Consider a 2×2 lattice
- Probe each quadrant...
- If 3 fail, POI/s *must* be in final
- Continue recursively!
- After 3 probes... distance halved!
- $P(n) \leq {\color{red}3}\lceil \log n \rceil$
- Total responses?

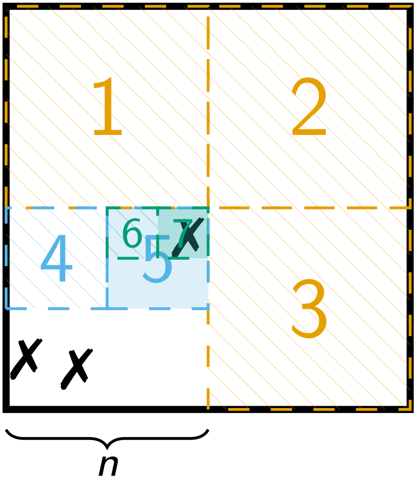


Figure 6: Quadrant Search

Rectilinear Strategies: Quadrant Algorithm

- Consider a 2×2 lattice
- Probe each quadrant...
- If 3 fail, POI/s *must* be in final
- Continue recursively!
- After 3 probes... distance halved!
- $P(n) \leq 3\lceil \log n \rceil$
- Total responses?
- At most one per layer...

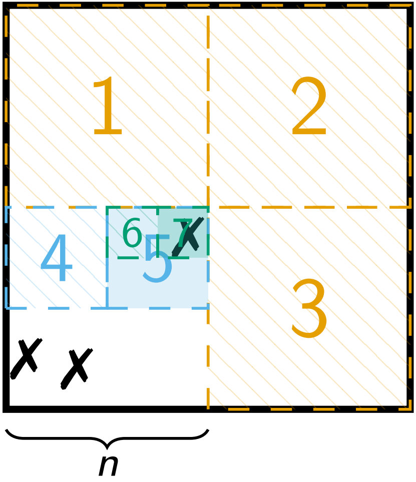


Figure 6: Quadrant Search

Rectilinear Strategies: Quadrant Algorithm

- Consider a 2×2 lattice
- Probe each quadrant...
- If 3 fail, POI/s *must* be in final
- Continue recursively!
- After 3 probes... distance halved!
- $P(n) \leq 3\lceil \log n \rceil$
- Total responses?
- At most one per layer...
- $R(n) \leq \lceil \log n \rceil$

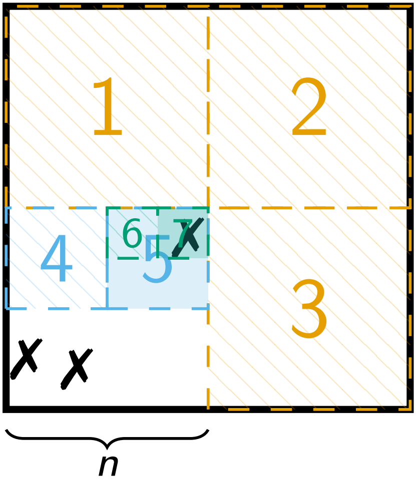


Figure 6: Quadrant Search

Rectilinear Strategies: Quadrant Algorithm

- Consider a 2×2 lattice
- Probe each quadrant...
- If 3 fail, POI/s *must* be in final
- Continue recursively!
- After 3 probes... distance halved!
- $P(n) \leq 3\lceil \log n \rceil$
- Total responses?
- At most one per layer...
- $R(n) \leq \lceil \log n \rceil$
- Distance traveled?

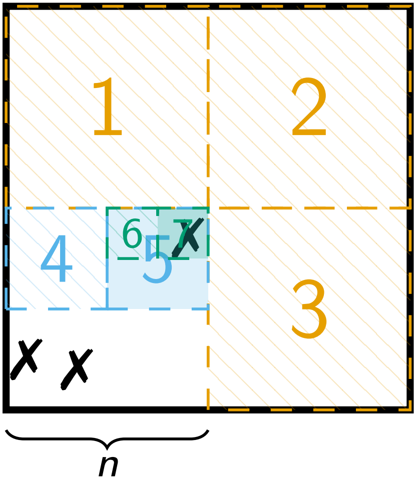


Figure 6: Quadrant Search

Rectilinear Strategies: Quadrant Algorithm

- Consider a 2×2 lattice
- Probe each quadrant...
- If 3 fail, POI/s *must* be in final
- Continue recursively!
- After 3 probes... distance halved!
- $P(n) \leq 3\lceil \log n \rceil$
- Total responses?
- At most one per layer...
- $R(n) \leq \lceil \log n \rceil$
- Distance traveled?
- $D(n) \leq 6n$

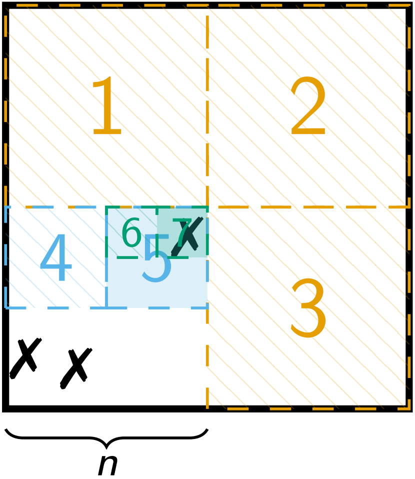


Figure 6: Quadrant Search

2D Domino Algorithm

- Quadrant: $P(n) \leq 3\lceil \log n \rceil$

2D Domino Algorithm

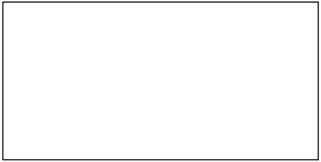
- Quadrant: $P(n) \leq 3\lceil \log n \rceil$
- Optimal is $2\lceil \log n \rceil$!

2D Domino Algorithm

- Quadrant: $P(n) \leq 3\lceil \log n \rceil$ – can we do better?
- Optimal is $2\lceil \log n \rceil$!

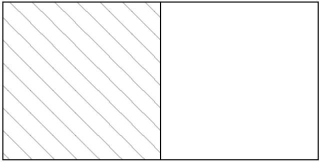
2D Domino Algorithm

- Quadrant: $P(n) \leq 3\lceil \log n \rceil$ – can we do better?
- Optimal is $2\lceil \log n \rceil$!
- Consider a $2n \times n$ '2-domino' configuration



2D Domino Algorithm

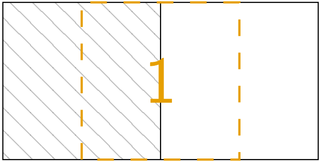
- Quadrant: $P(n) \leq 3\lceil \log n \rceil$ – can we do better?
- Optimal is $2\lceil \log n \rceil$!
- Consider a $2n \times n$ '2-domino' configuration
- One side known to be *empty*, other known to contain POI/s



2D Domino Algorithm

- Quadrant: $P(n) \leq 3\lceil \log n \rceil$ – can we do better?
- Optimal is $2\lceil \log n \rceil$!
- Consider a $2n \times n$ '2-domino' configuration
- One side known to be *empty*, other known to contain POI/s

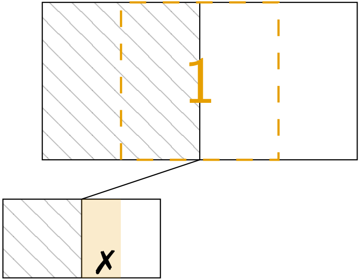
1. Probe center



2D Domino Algorithm

- Quadrant: $P(n) \leq 3\lceil \log n \rceil$ – can we do better?
- Optimal is $2\lceil \log n \rceil$!
- Consider a $2n \times n$ '2-domino' configuration
- One side known to be *empty*, other known to contain POI/s

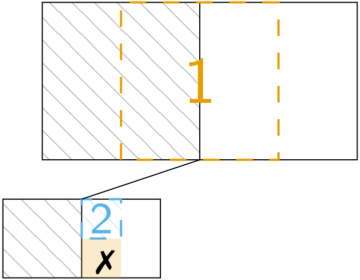
1. Probe center
2. Probe remaining quadrant
   a. If 1st probe succeeded...



2D Domino Algorithm

- Quadrant: $P(n) \leq 3\lceil \log n \rceil$ – can we do better?
- Optimal is $2\lceil \log n \rceil$!
- Consider a $2n \times n$ '2-domino' configuration
- One side known to be *empty*, other known to contain POI/s

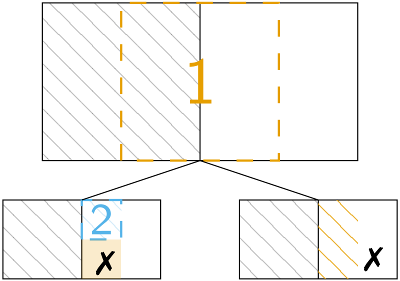
1. Probe center
2. Probe remaining quadrant
   a. If 1st probe succeeded...



2D Domino Algorithm

- Quadrant: $P(n) \leq 3\lceil\log n\rceil$ – can we do better?
- Optimal is $2\lceil\log n\rceil$!
- Consider a $2n \times n$ '2-domino' configuration
- One side known to be *empty*, other known to contain POI/s

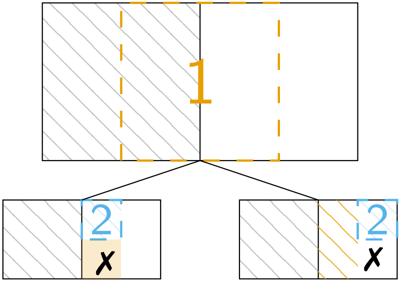
1. Probe center
2. Probe remaining quadrant
   1. If 1st probe succeeded...
   2. If 1st probe failed...



2D Domino Algorithm

- Quadrant: $P(n) \leq \textcolor{red}{3}\lceil \log n \rceil$ – can we do better?
- Optimal is $\textcolor{red}{2}\lceil \log n \rceil$!
- Consider a $2n \times n$ '2-domino' configuration
- One side known to be *empty*, other known to contain POI/s

1. Probe center
2. Probe remaining quadrant
   a. If 1st probe succeeded...
   b. If 1st probe failed...

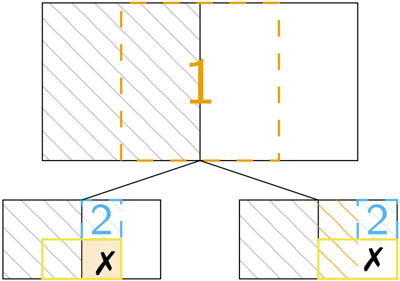


2D Domino Algorithm

- Quadrant: $P(n) \leq 3\lceil \log n \rceil$ – can we do better?
- Optimal is $2\lceil \log n \rceil$!
- Consider a $2n \times n$ '2-domino' configuration
- One side known to be *empty*, other known to contain POI/s

1. Probe center
2. Probe remaining quadrant
   a. If 1st probe succeeded...
   b. If 1st probe failed...

- Either way... $n \times n/2$ 'domino'

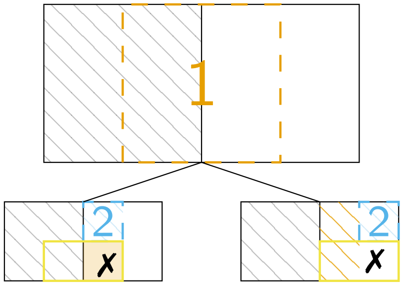


2D Domino Algorithm

- Quadrant: $P(n) \leq 3\lceil\log n\rceil$ – can we do better?
- Optimal is $2\lceil\log n\rceil$!
- Consider a $2n \times n$ '2-domino' configuration
- One side known to be *empty*, other known to contain POI/s

1. Probe center
2. Probe remaining quadrant
   a. If 1st probe succeeded...
   b. If 1st probe failed...

- Either way... $n \times n/2$ 'domino'
- Two probes $\rightarrow$ halve dimensions

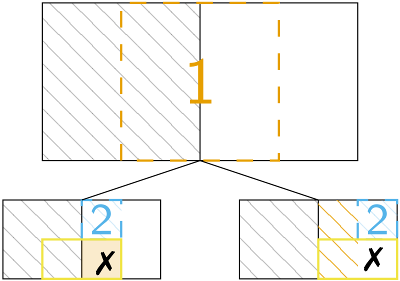


2D Domino Algorithm

- Quadrant: $P(n) \leq 3\lceil \log n \rceil$ – can we do better?
- Optimal is $2\lceil \log n \rceil$!
- Consider a $2n \times n$ '2-domino' configuration
- One side known to be *empty*, other known to contain POI/s

1. Probe center
2. Probe remaining quadrant
   a. If 1st probe succeeded...
   b. If 1st probe failed...

- Either way... $n \times n/2$ 'domino'
- Two probes $\rightarrow$ halve dimensions
- $P(n) = 2\lceil \log n \rceil$

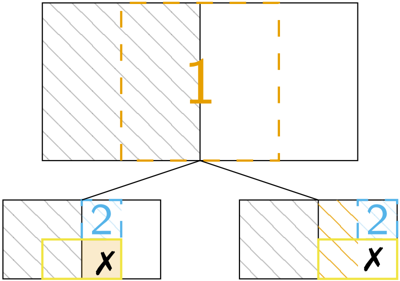


2D Domino Algorithm

- Quadrant: $P(n) \leq 3\lceil \log n \rceil$ – can we do better?
- Optimal is $2\lceil \log n \rceil$!
- Consider a $2n \times n$ '2-domino' configuration
- One side known to be *empty*, other known to contain POI/s

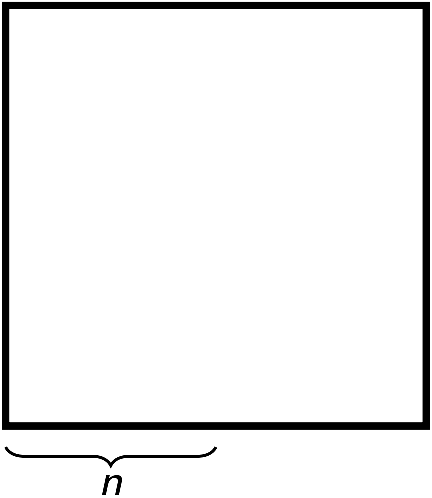
1. Probe center
2. Probe remaining quadrant
   a. If 1st probe succeeded...
   b. If 1st probe failed...

- Either way... $n \times n/2$ 'domino'
- Two probes $\rightarrow$ halve dimensions
- $P(n) = 2\lceil \log n \rceil$
- How to reach domino?



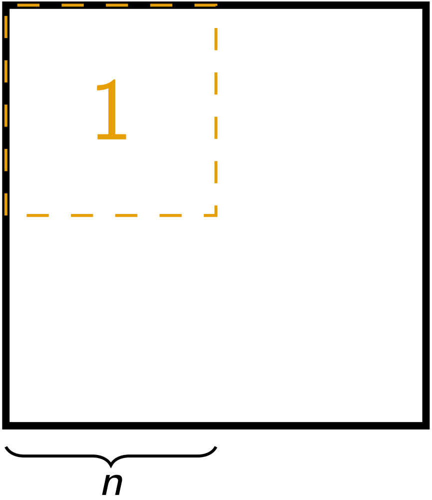
Rectilinear Strategies: 2D Domino Algorithm – cont'd

- How to reach domino?



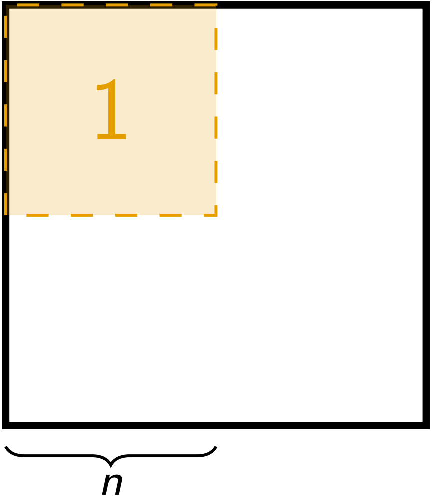
Rectilinear Strategies: 2D Domino Algorithm – cont'd

- How to reach domino?
- Start quadrant search...



Rectilinear Strategies: 2D Domino Algorithm – cont'd

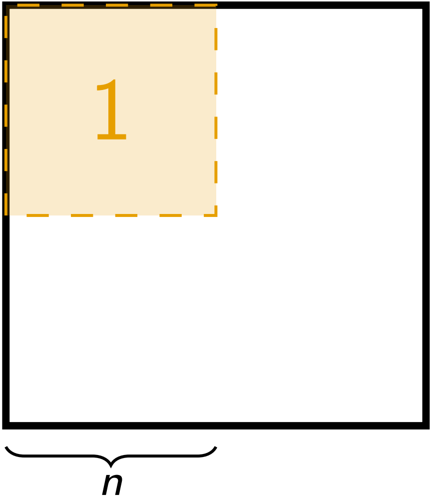
- How to reach domino?
- Start quadrant search...
- If first probe succeeds...



Rectilinear Strategies: 2D Domino Algorithm – cont'd

- How to reach domino?
- Start quadrant search...

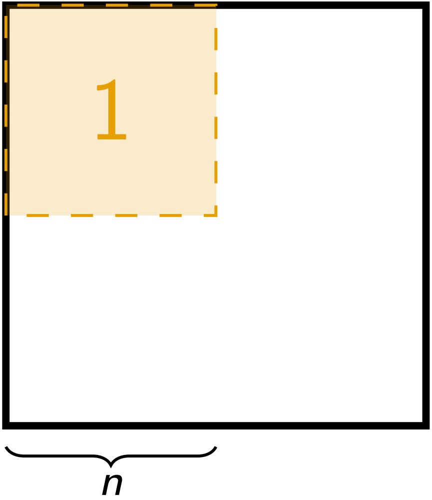
- If first probe succeeds...
- Reduce area by factor of 4!



Rectilinear Strategies: 2D Domino Algorithm – cont'd

- How to reach domino?
- Start quadrant search...

- If first probe succeeds...
- Reduce area by factor of 4!
- Just recurse and ☺

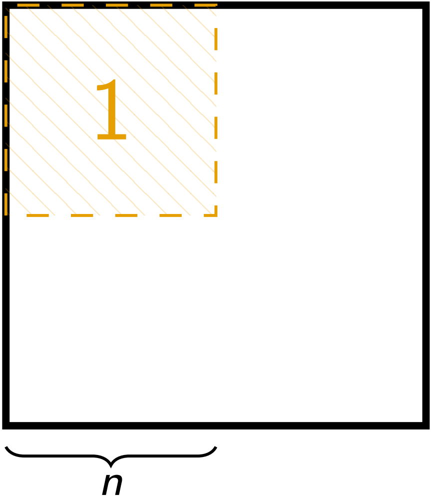


Rectilinear Strategies: 2D Domino Algorithm – cont'd

- How to reach domino?
- Start quadrant search...

- If first probe succeeds...
- Reduce area by factor of 4!
- Just recurse and ☺

- If first probe fails...

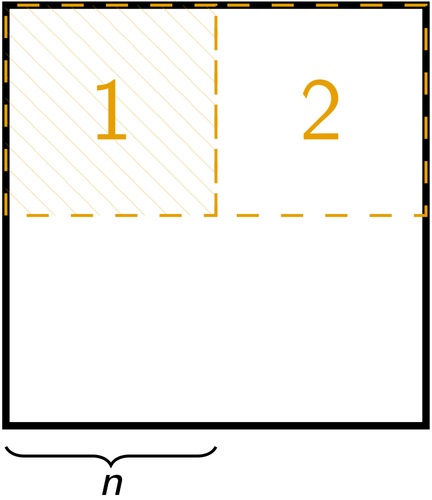


Rectilinear Strategies: 2D Domino Algorithm – cont'd

- How to reach domino?
- Start quadrant search...

- If first probe succeeds...
- Reduce area by factor of 4!
- Just recurse and ☺

- If first probe fails...
- 2-domino guaranteed!



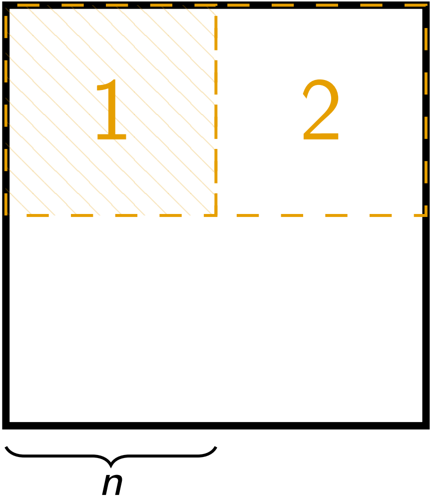
Rectilinear Strategies: 2D Domino Algorithm – cont'd

- How to reach domino?
- Start quadrant search...

- If first probe succeeds...
- Reduce area by factor of 4!
- Just recurse and ☺

- If first probe fails...
- 2-domino guaranteed!

- $P(n) \leq 2\lceil \log n \rceil + 1$



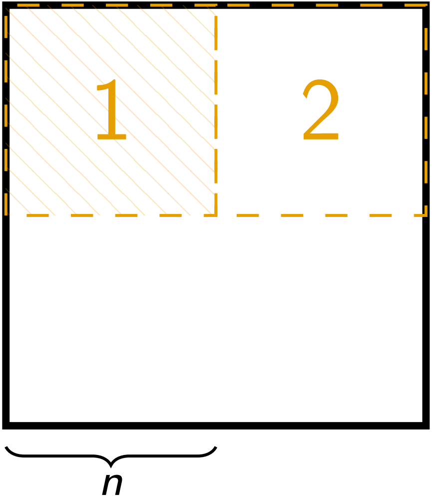
Rectilinear Strategies: 2D Domino Algorithm – cont'd

- How to reach domino?
- Start quadrant search...

- If first probe succeeds...
- Reduce area by factor of 4!
- Just recurse and ☺

- If first probe fails...
- 2-domino guaranteed!

- $P(n) \leq 2\lceil \log n \rceil + 1$ ✓



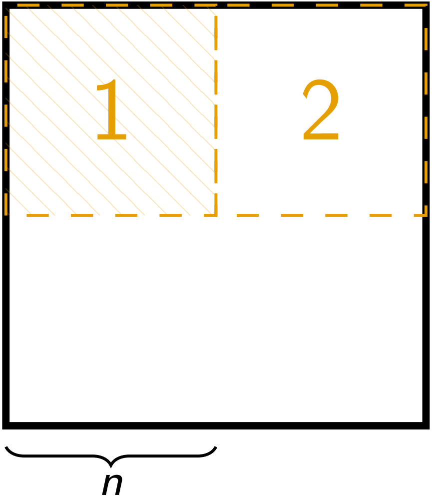
Rectilinear Strategies: 2D Domino Algorithm – cont'd

- How to reach domino?
- Start quadrant search...

- If first probe succeeds...
- Reduce area by factor of 4!
- Just recurse and ☺

- If first probe fails...
- 2-domino guaranteed!

- $P(n) \leq 2\lceil \log n \rceil + 1$ ✓
- $R(n) = 2\lceil \log n \rceil - 1$



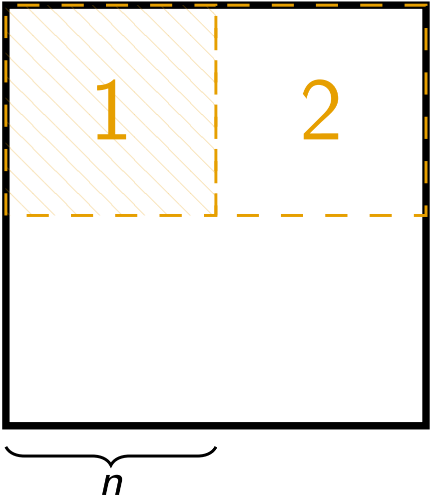
Rectilinear Strategies: 2D Domino Algorithm – cont'd

- How to reach domino?
- Start quadrant search...

- If first probe succeeds...
- Reduce area by factor of 4!
- Just recurse and ☺

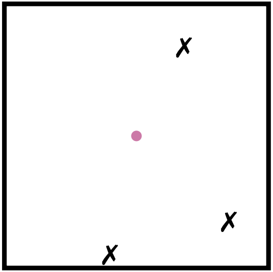
- If first probe fails...
- 2-domino guaranteed!

- $P(n) \leq 2\lceil \log n \rceil + 1$ ✓
- $R(n) = 2\lceil \log n \rceil - 1$
- $D(n) < 6n$



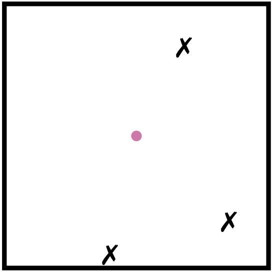
Rectilinear Strategies: Central Binary Search (CBS)

- Recall: $D(n) \in \mathcal{O}(n)$



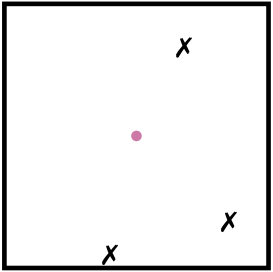
Rectilinear Strategies: Central Binary Search (CBS)

- Recall: $D(n) \in \mathcal{O}(n)$
- If POI is nearby?



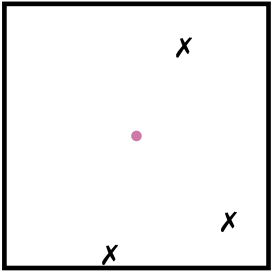
Rectilinear Strategies: Central Binary Search (CBS)

- Recall: $D(n) \in \mathcal{O}(n)$
- If POI is nearby? still $\mathcal{O}(n)$



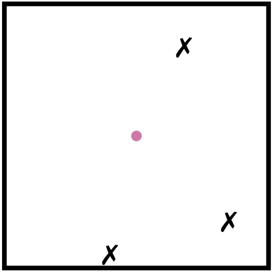
Rectilinear Strategies: Central Binary Search (CBS)

- Recall: $D(n) \in \mathcal{O}(n)$
- If POI is nearby? still $\mathcal{O}(n)$
- Nearest POI at distance δ_{min}



Rectilinear Strategies: Central Binary Search (CBS)

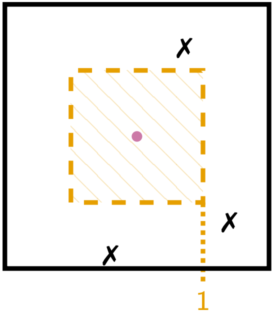
- Recall: $D(n) \in \mathcal{O}(n)$
- If POI is nearby? still $\mathcal{O}(n)$
- Nearest POI at distance δ_{min}
- Can we do $\mathcal{O}(\delta_{\text{min}})$?



Rectilinear Strategies: Central Binary Search (CBS)

- Recall: $D(n) \in \mathcal{O}(n)$
- If POI is nearby? still $\mathcal{O}(n)$
- Nearest POI at distance $\delta_{\min}$
- Can we do $\mathcal{O}(\delta_{\min})$?

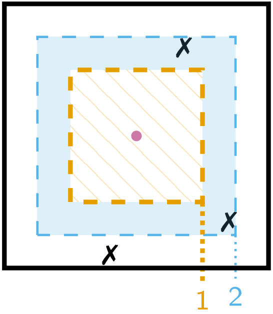
- Perform binary search.



Rectilinear Strategies: Central Binary Search (CBS)

- Recall: $D(n) \in \mathcal{O}(n)$
- If POI is nearby? still $\mathcal{O}(n)$
- Nearest POI at distance $\delta_{\min}$
- Can we do $\mathcal{O}(\delta_{\min})$?

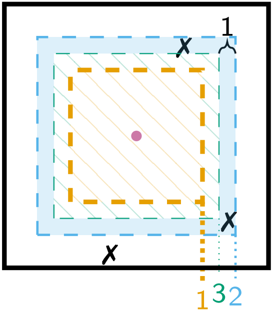
- Perform binary search..



Rectilinear Strategies: Central Binary Search (CBS)

- Recall: $D(n) \in \mathcal{O}(n)$
- If POI is nearby? still $\mathcal{O}(n)$
- Nearest POI at distance $\delta_{\min}$
- Can we do $\mathcal{O}(\delta_{\min})$?

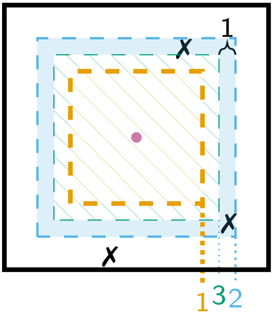
- Perform binary search...
- Stop when width-1 shell*



Rectilinear Strategies: Central Binary Search (CBS)

- Recall: $D(n) \in \mathcal{O}(n)$
- If POI is nearby? still $\mathcal{O}(n)$
- Nearest POI at distance $\delta_{\min}$
- Can we do $\mathcal{O}(\delta_{\min})$?

- Perform binary search...
- Stop when width-1 shell*
- $\lceil \log n \rceil$ probes, 0 distance

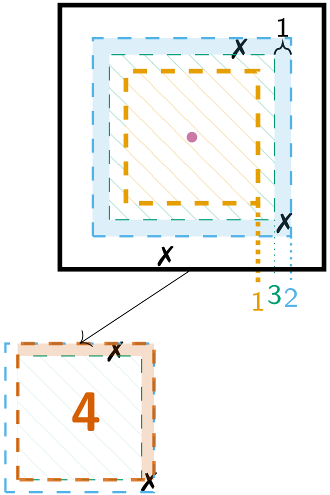


Rectilinear Strategies: Central Binary Search (CBS)

- Recall: $D(n) \in \mathcal{O}(n)$
- If POI is nearby? still $\mathcal{O}(n)$
- Nearest POI at distance $\delta_{\min}$
- Can we do $\mathcal{O}(\delta_{\min})$?

- Perform binary search...
- Stop when width-1 shell*
- $\lceil \log n \rceil$ probes, 0 distance

- Determine edge containing POI.

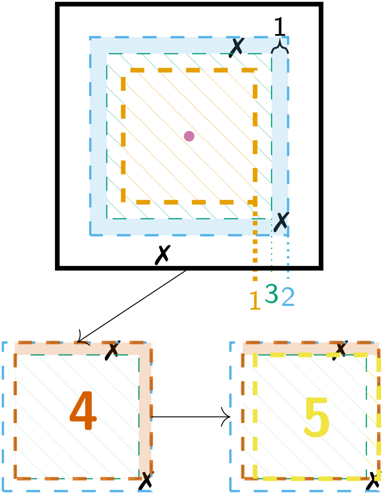


Rectilinear Strategies: Central Binary Search (CBS)

- Recall: $D(n) \in \mathcal{O}(n)$
- If POI is nearby? still $\mathcal{O}(n)$
- Nearest POI at distance $\delta_{\min}$
- Can we do $\mathcal{O}(\delta_{\min})$?

- Perform binary search...
- Stop when width-1 shell*
- $\lceil \log n \rceil$ probes, 0 distance

- Determine edge containing POI..
- 2 probes

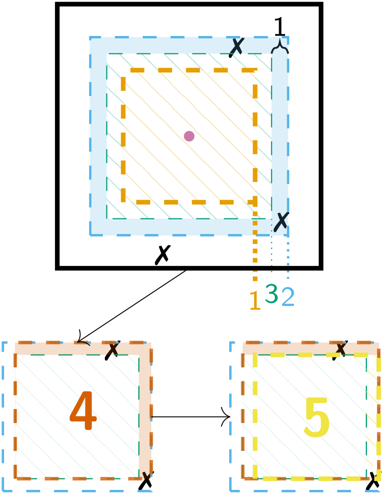


Rectilinear Strategies: Central Binary Search (CBS)

- Recall: $D(n) \in \mathcal{O}(n)$
- If POI is nearby? still $\mathcal{O}(n)$
- Nearest POI at distance $\delta_{\min}$
- Can we do $\mathcal{O}(\delta_{\min})$?

- Perform binary search...
- Stop when width-1 shell*
- $\lceil \log n \rceil$ probes, 0 distance

- Determine edge containing POI..
- 2 probes – near origin!

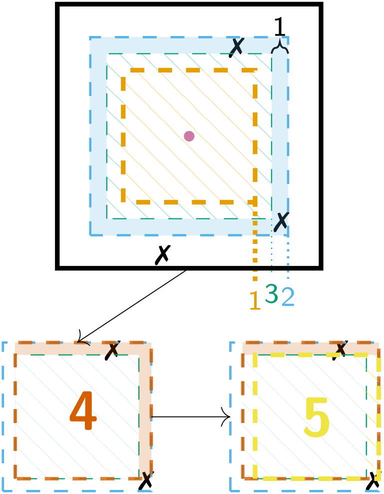


Rectilinear Strategies: Central Binary Search (CBS)

- Recall: $D(n) \in \mathcal{O}(n)$
- If POI is nearby? still $\mathcal{O}(n)$
- Nearest POI at distance $\delta_{\min}$
- Can we do $\mathcal{O}(\delta_{\min})$?

- Perform binary search...
- Stop when width-1 shell*
- $\lceil \log n \rceil$ probes, 0 distance

- Determine edge containing POI..
- 2 probes – near origin!
- $\mathcal{O}(1)$ distance



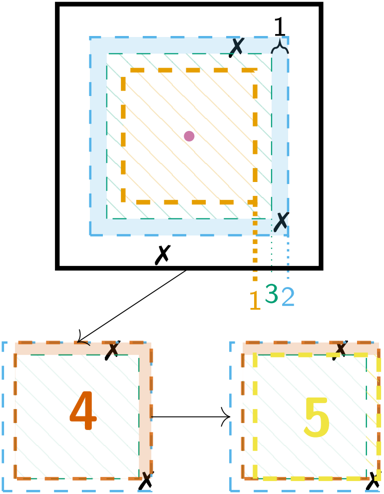
Rectilinear Strategies: Central Binary Search (CBS)

- Recall: $D(n) \in \mathcal{O}(n)$
- If POI is nearby? still $\mathcal{O}(n)$
- Nearest POI at distance $\delta_{\min}$
- Can we do $\mathcal{O}(\delta_{\min})$?

- Perform binary search...
- Stop when width-1 shell*
- $\lceil \log n \rceil$ probes, 0 distance

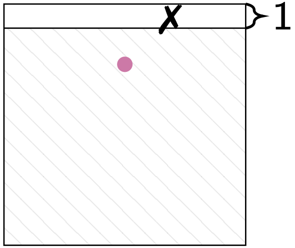
- Determine edge containing POI..
- 2 probes – near origin!
- $\mathcal{O}(1)$ distance

- What now?



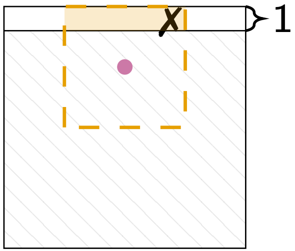
Rectilinear Strategies: CBS – 2nd Phase

- What now?



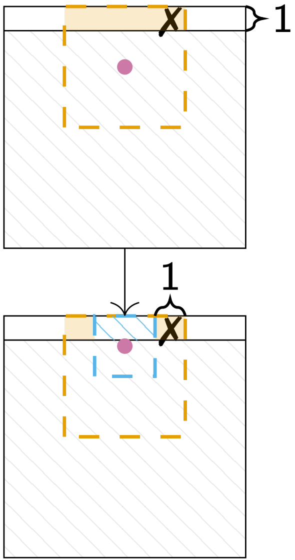
Rectilinear Strategies: CBS – 2nd Phase

- What now?
- Another binary search!



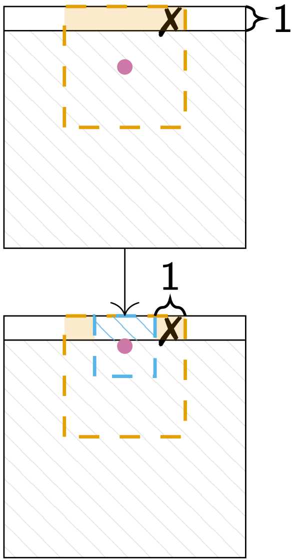
Rectilinear Strategies: CBS – 2nd Phase

- What now?
- Another binary search!
- At most $\lceil \log n \rceil$ probes



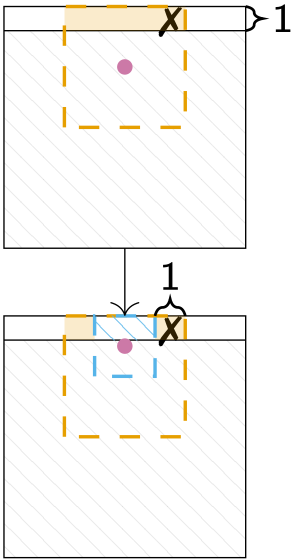
Rectilinear Strategies: CBS – 2nd Phase

- What now?
- Another binary search!
- At most $\lceil \log n \rceil$ probes
- Distance?



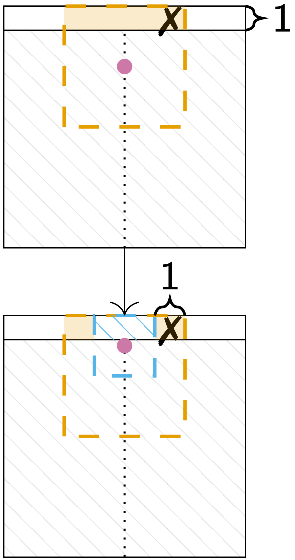
Rectilinear Strategies: CBS – 2nd Phase

- What now?
- Another binary search!
- At most $\lceil \log n \rceil$ probes
- Distance? *not* stationary!



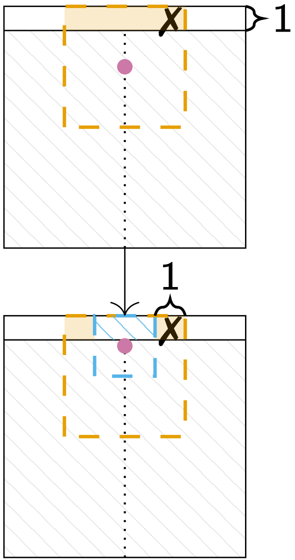
Rectilinear Strategies: CBS – 2nd Phase

- What now?
- Another binary search!
- At most $\lceil \log n \rceil$ probes
- Distance? *not* stationary!
- Always moves along axis...



Rectilinear Strategies: CBS – 2nd Phase

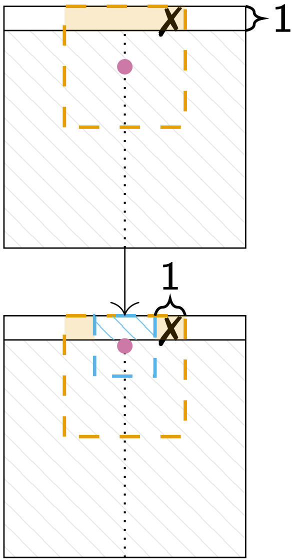
- What now?
- Another binary search!
- At most $\lceil \log n \rceil$ probes
- Distance? *not* stationary!
- Always moves along axis...
- First probe, $\leq \delta_{\min}/2$



Rectilinear Strategies: CBS – 2nd Phase

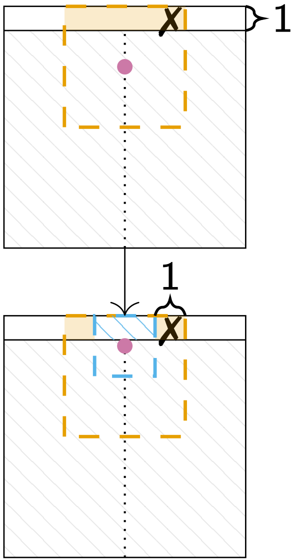
- What now?
- Another binary search!
- At most $\lceil \log n \rceil$ probes

- Distance? *not* stationary!
- Always moves along axis...
- First probe, $\leq \delta_{\min}/2$
- Second, $\leq \delta_{\min}/4$



Rectilinear Strategies: CBS – 2nd Phase

- What now?
- Another binary search!
- At most $\lceil \log n \rceil$ probes
- Distance? *not* stationary!
- Always moves along axis...
- First probe, $\leq \delta_{\min}/2$
- Second, $\leq \delta_{\min}/4 \rightarrow \Sigma \ldots \leq \delta_{\min}$

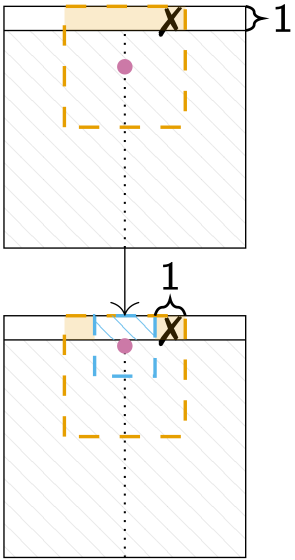


Rectilinear Strategies: CBS – 2nd Phase

- What now?
- Another binary search!
- At most $\lceil \log n \rceil$ probes

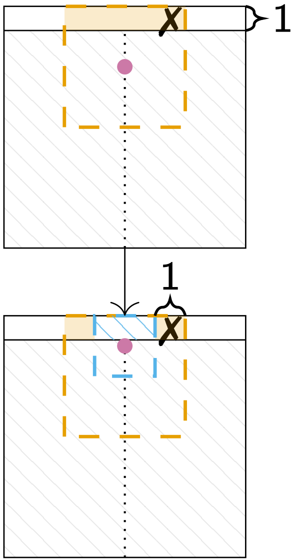
- Distance? *not* stationary!
- Always moves along axis. . .
- First probe, $\leq \delta_{\min}/2$
- Second, $\leq \delta_{\min}/4 \to \Sigma \ldots \leq \delta_{\min}$

- 2 remaining 1×1 regions



Rectilinear Strategies: CBS – 2nd Phase

- What now?
- Another binary search!
- At most $\lceil \log n \rceil$ probes
- Distance? *not* stationary!
- Always moves along axis...
- First probe, $\leq \delta_{\text{min}}/2$
- Second, $\leq \delta_{\text{min}}/4 \rightarrow \Sigma ... \leq \delta_{\text{min}}$
- 2 remaining 1×1 regions
- 1 more probe, $\mathcal{O}(1)$ distance

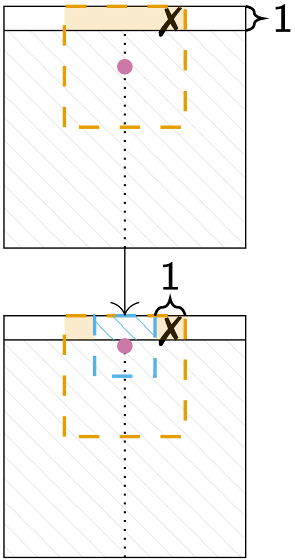


Rectilinear Strategies: CBS – 2nd Phase

- What now?
- Another binary search!
- At most $\lceil \log n \rceil$ probes

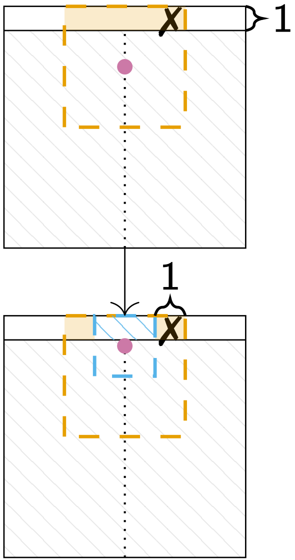
- Distance? *not* stationary!
- Always moves along axis...
- First probe, $\leq \delta_{\min}/2$
- Second, $\leq \delta_{\min}/4 \to \Sigma\ldots \leq \delta_{\min}$

- 2 remaining 1×1 regions
- 1 more probe, $\mathcal{O}(1)$ distance
- $\leq \delta_{\min}$ distance to reach POI...



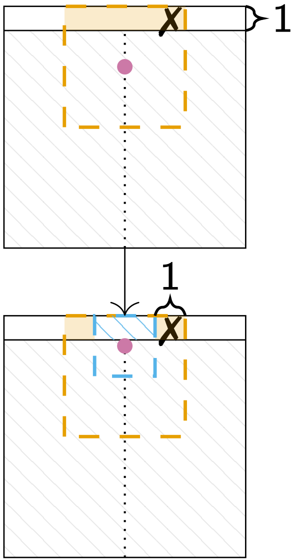
Rectilinear Strategies: CBS – 2nd Phase

- What now?
- Another binary search!
- At most $\lceil \log n \rceil$ probes
- Distance? *not* stationary!
- Always moves along axis...
- First probe, $\leq \delta_{\min}/2$
- Second, $\leq \delta_{\min}/4 \rightarrow \Sigma ... \leq \delta_{\min}$
- 2 remaining 1×1 regions
- 1 more probe, $\mathcal{O}(1)$ distance
- $\leq \delta_{\min}$ distance to reach POI...
- $D(n) \leq 2\delta_{\min} + \mathcal{O}(1)$



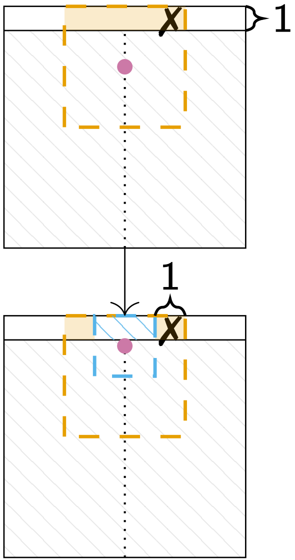
Rectilinear Strategies: CBS – 2nd Phase

- What now?
- Another binary search!
- At most $\lceil \log n \rceil$ probes
- Distance? *not* stationary!
- Always moves along axis...
- First probe, $\leq \delta_{\min}/2$
- Second, $\leq \delta_{\min}/4 \rightarrow \Sigma \ldots \leq \delta_{\min}$
- 2 remaining 1×1 regions
- 1 more probe, $\mathcal{O}(1)$ distance
- $\leq \delta_{\min}$ distance to reach POI...
- $D(n) \leq 2\delta_{\min} + \mathcal{O}(1)$ ✓



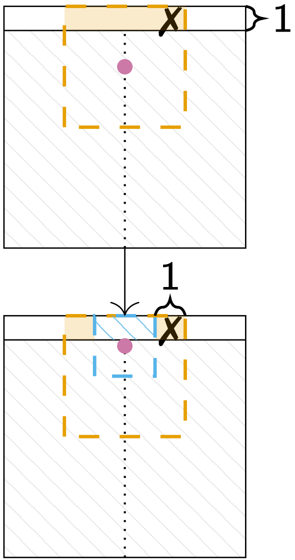
Rectilinear Strategies: CBS – 2nd Phase

- What now?
- Another binary search!
- At most $\lceil \log n \rceil$ probes
- Distance? *not* stationary!
- Always moves along axis...
- First probe, $\leq \delta_{\text{min}}/2$
- Second, $\leq \delta_{\text{min}}/4 \to \Sigma \ldots \leq \delta_{\text{min}}$
- 2 remaining 1×1 regions
- 1 more probe, $\mathcal{O}(1)$ distance
- $\leq \delta_{\text{min}}$ distance to reach POI...
- $D(n) \leq 2\delta_{\text{min}} + \mathcal{O}(1)$ ✓
- $P(n) \leq 2\lceil \log n \rceil + 1$

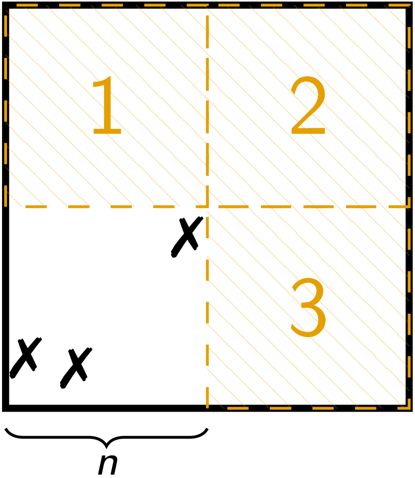


Rectilinear Strategies: CBS – 2nd Phase

- What now?
- Another binary search!
- At most $\lceil \log n \rceil$ probes
- Distance? *not* stationary!
- Always moves along axis. . .
- First probe, $\leq \delta_{\text{min}}/2$
- Second, $\leq \delta_{\text{min}}/4 \rightarrow \Sigma\ldots \leq \delta_{\text{min}}$
- 2 remaining 1×1 regions
- 1 more probe, $\mathcal{O}(1)$ distance
- $\leq \delta_{\text{min}}$ distance to reach POI...
- $D(n) \leq 2\delta_{\text{min}} + \mathcal{O}(1)$ ✓
- $P(n) \leq 2\lceil \log n \rceil + 1$ ✓

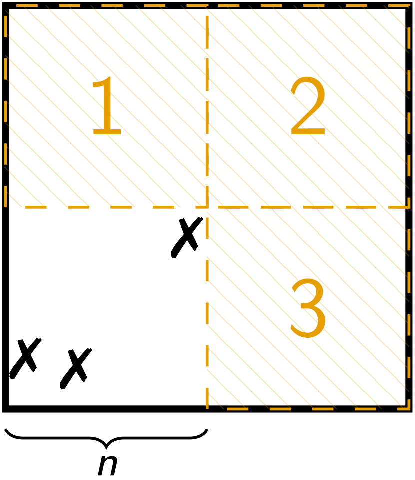


- Is 2D the limit?



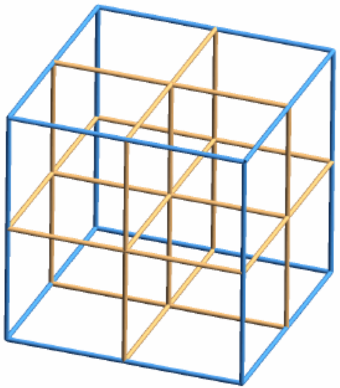
Rectilinear Strategies: Higher dimensions?

- Is 2D the limit?
- Recall: Quadrant algorithm



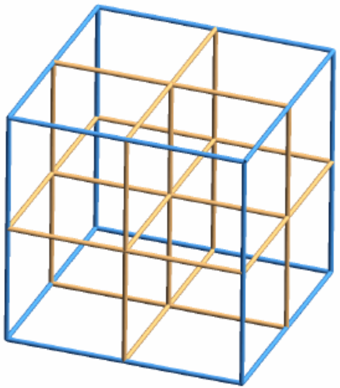
Rectilinear Strategies: Higher dimensions?

- Is 2D the limit?
- Recall: Quadrant algorithm
- Octant algorithm?



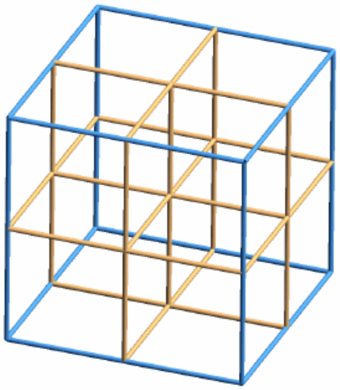
Rectilinear Strategies: Higher dimensions?

- Is 2D the limit?
- Recall: Quadrant algorithm
- Octant algorithm?
- $P(n) \leq {\color{red}7}\lceil \log n \rceil$



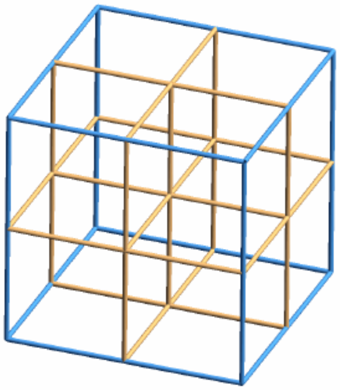
Rectilinear Strategies: Higher dimensions?

- Is 2D the limit?
- Recall: Quadrant algorithm
- Octant algorithm?
- $P(n) \leq 7\lceil \log n \rceil$
- # responses?



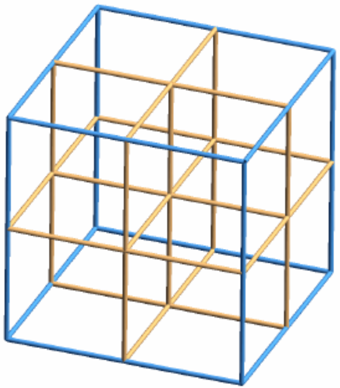
Rectilinear Strategies: Higher dimensions?

- Is 2D the limit?
- Recall: Quadrant algorithm
- Octant algorithm?
- $P(n) \leq 7\lceil \log n \rceil$
- # responses? still $\lceil \log n \rceil$!



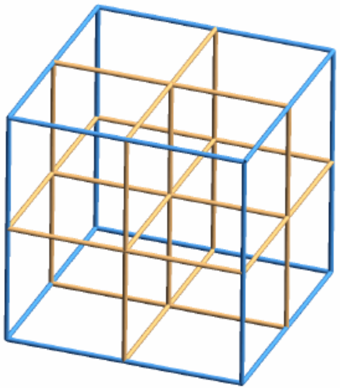
Rectilinear Strategies: Higher dimensions?

- Is 2D the limit?
- Recall: Quadrant algorithm
- Octant algorithm?
- $P(n) \leq 7\lceil \log n \rceil$
- # responses? still $\lceil \log n \rceil$!
- Further?



Rectilinear Strategies: Higher dimensions?

- Is 2D the limit?
- Recall: Quadrant algorithm
- Octant algorithm?
- $P(n) \leq 7\lceil \log n \rceil$
- # responses? still $\lceil \log n \rceil$!
- Further?
- kD hypercube $\rightarrow 2^k$ 'orthants'

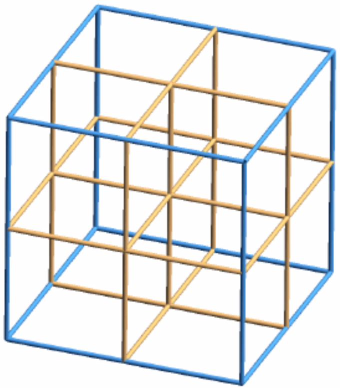


Rectilinear Strategies: Higher dimensions?

- Is 2D the limit?

- Recall: Quadrant algorithm
- Octant algorithm?
- $P(n) \leq 7\lceil \log n \rceil$
- $\#$ responses? still $\lceil \log n \rceil$!

- Further?
- kD hypercube $\rightarrow 2^k$ 'orthants'
- Orthant algorithm
- $P(n) \leq (2^k - 1)\lceil \log n \rceil$



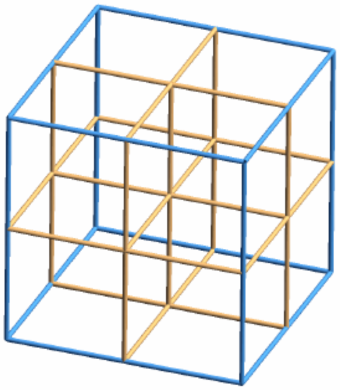
Rectilinear Strategies: Higher dimensions?

- Is 2D the limit?

- Recall: Quadrant algorithm
- Octant algorithm?
- $P(n) \leq 7\lceil \log n \rceil$
- $\#$ responses? still $\lceil \log n \rceil$!

- Further?
- kD hypercube $\rightarrow 2^k$ 'orthants'

- Orthant algorithm
- $P(n) \leq (2^k - 1)\lceil \log n \rceil$
- Terrible distance $> 2^k n$



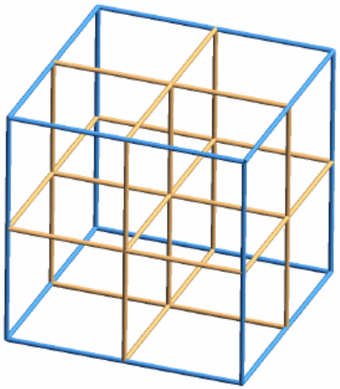
Rectilinear Strategies: Higher dimensions?

- Is 2D the limit?

- Recall: Quadrant algorithm
- Octant algorithm?
- $P(n) \leq 7\lceil \log n \rceil$
- $\#$ responses? still $\lceil \log n \rceil$!

- Further?
- kD hypercube $\rightarrow 2^k$ 'orthants'

- Orthant algorithm
- $P(n) \leq (2^k - 1)\lceil \log n \rceil$
- Terrible distance $> 2^k n$
- Excellent responses, $\leq \lceil \log n \rceil$

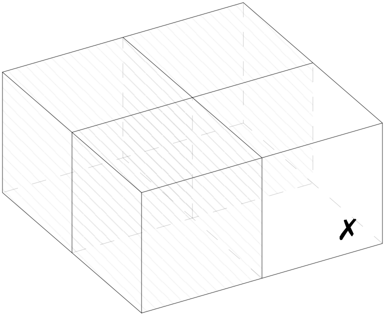


Rectilinear Strategies: 3D Domino

- How about 3D domino?

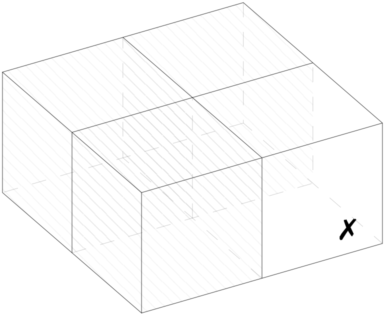
Rectilinear Strategies: 3D Domino

- How about 3D domino?
- Consider $2n \times 2n \times n$ '4-domino'



Rectilinear Strategies: 3D Domino

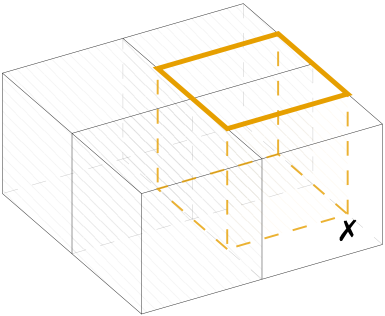
- How about 3D domino?
- Consider $2n \times 2n \times n$ '4-domino'
- 3 empty, 1 contains POI/s



Rectilinear Strategies: 3D Domino

- How about 3D domino?

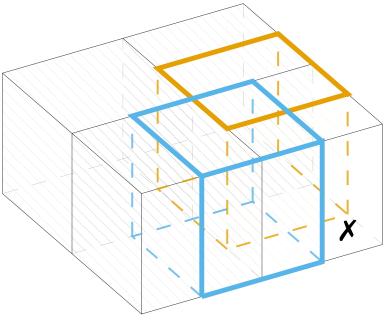
- Consider $2n \times 2n \times n$ '4-domino'
- 3 empty, 1 contains POI/s
- After 3 probes.



Rectilinear Strategies: 3D Domino

- How about 3D domino?

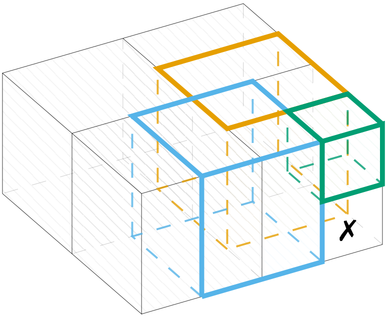
- Consider $2n \times 2n \times n$ '4-domino'
- 3 empty, 1 contains POI/s
- After 3 probes..



Rectilinear Strategies: 3D Domino

- How about 3D domino?

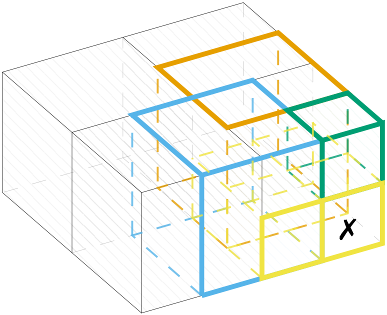
- Consider $2n \times 2n \times n$ '4-domino'
- 3 empty, 1 contains POI/s
- After 3 probes...



Rectilinear Strategies: 3D Domino

- How about 3D domino?

- Consider $2n \times 2n \times n$ '4-domino'
- 3 empty, 1 contains POI/s
- After 3 probes...
- $1/8$ volume

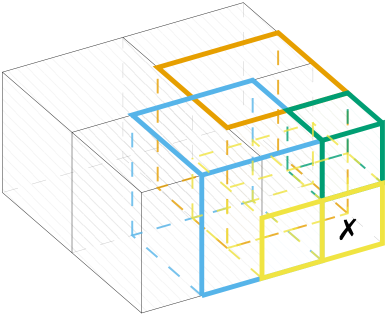


Rectilinear Strategies: 3D Domino

- How about 3D domino?

- Consider $2n \times 2n \times n$ '4-domino'
- 3 empty, 1 contains POI/s
- After 3 probes...
- $1/8$ volume

- Probes halve volume

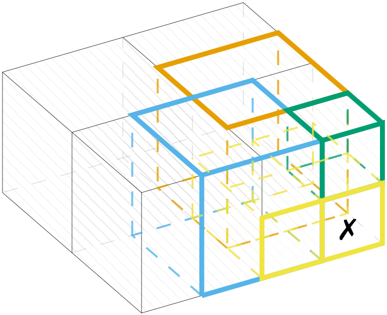


Rectilinear Strategies: 3D Domino

- How about 3D domino?

- Consider $2n \times 2n \times n$ '4-domino'
- 3 empty, 1 contains POI/s
- After 3 probes...
- $1/8$ volume

- Probes halve volume – optimal

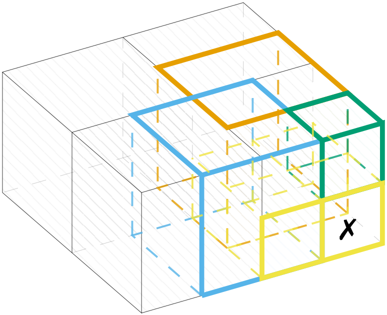


Rectilinear Strategies: 3D Domino

- How about 3D domino?

- Consider $2n \times 2n \times n$ '4-domino'
- 3 empty, 1 contains POI/s
- After 3 probes...
- $1/8$ volume

- Probes halve volume – optimal
- $P(n) = 3\lceil \log n \rceil$



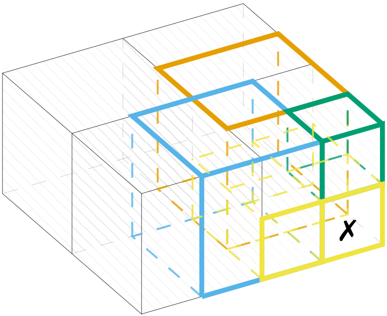
Rectilinear Strategies: 3D Domino

- How about 3D domino?

- Consider $2n \times 2n \times n$ '4-domino'
- 3 empty, 1 contains POI/s
- After 3 probes...
- $1/8$ volume

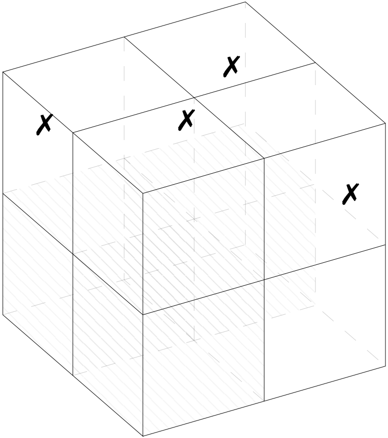
- Probes halve volume – optimal
- $P(n) = 3\lceil \log n \rceil$

- How to reach 4-domino?



Rectilinear Strategies: 3D Domino

- How about 3D domino?
- Consider $2n \times 2n \times n$ '4-domino'
- 3 empty, 1 contains POI/s
- After 3 probes...
- $1/8$ volume
- Probes halve volume – optimal
- $P(n) = 3\lceil \log n \rceil$
- How to reach 4-domino?
- May take more than 4 probes...



Rectilinear Strategies: 3D Domino

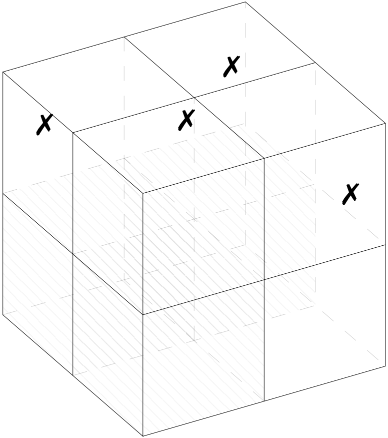
- How about 3D domino?

- Consider $2n \times 2n \times n$ '4-domino'
- 3 empty, 1 contains POI/s
- After 3 probes...
- $1/8$ volume

- Probes halve volume – optimal
- $P(n) = 3\lceil \log n \rceil$

- How to reach 4-domino?

- May take more than 4 probes...
- If 4th probe always succeeds...



Rectilinear Strategies: 3D Domino

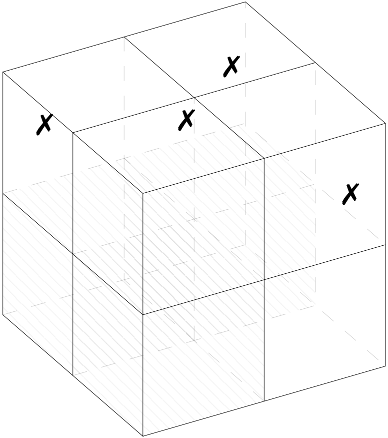
- How about 3D domino?

- Consider $2n \times 2n \times n$ '4-domino'
- 3 empty, 1 contains POI/s
- After 3 probes...
- $1/8$ volume

- Probes halve volume – optimal
- $P(n) = 3\lceil \log n \rceil$

- How to reach 4-domino?

- May take more than 4 probes...
- If 4th probe always succeeds...
- $P(n) \leq 4\lceil \log n \rceil$



Rectilinear Strategies: 3D Domino

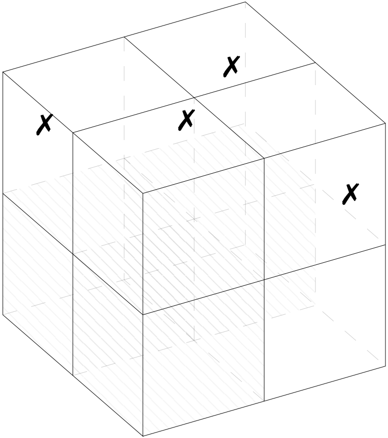
- How about 3D domino?

- Consider $2n \times 2n \times n$ '4-domino'
- 3 empty, 1 contains POI/s
- After 3 probes...
- $1/8$ volume

- Probes halve volume – optimal
- $P(n) = 3\lceil \log n \rceil$

- How to reach 4-domino?

- May take more than 4 probes...
- If 4th probe always succeeds...
- $P(n) \leq 4\lceil \log n \rceil$ – not optimal!

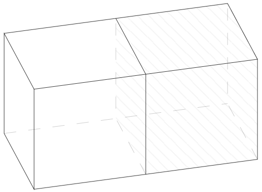


Rectilinear Strategies: 3D Domino – cont'd

- Idea: Intermediate configuration

Rectilinear Strategies: 3D Domino – cont'd

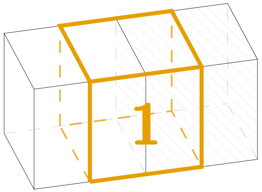
- Idea: Intermediate configuration
- 3D 2-domino: $2n \times n \times n$



Rectilinear Strategies: 3D Domino – cont'd

- Idea: Intermediate configuration
- 3D 2-domino: $2n \times n \times n$

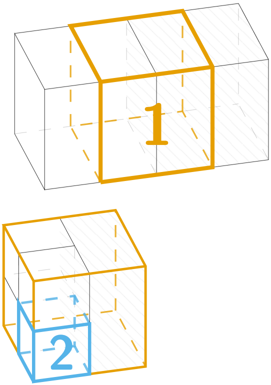
1. Probe center



Rectilinear Strategies: 3D Domino – cont'd

- Idea: Intermediate configuration
- 3D 2-domino: $2n \times n \times n$

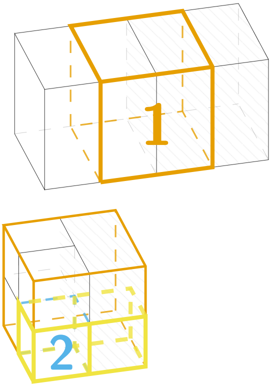
1. Probe center
2. Probe remaining 'quadrant'



Rectilinear Strategies: 3D Domino – cont'd

- Idea: Intermediate configuration
- 3D 2-domino: $2n \times n \times n$

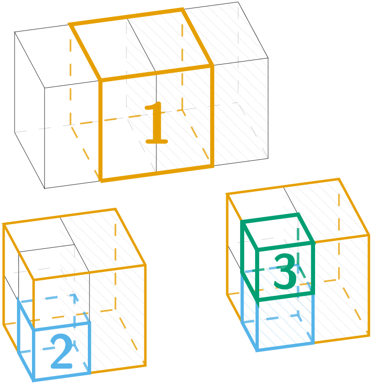
1. Probe center
2. Probe remaining 'quadrant'
    a. If success. . . 2-domino!



Rectilinear Strategies: 3D Domino – cont'd

- Idea: Intermediate configuration
- 3D 2-domino: $2n \times n \times n$

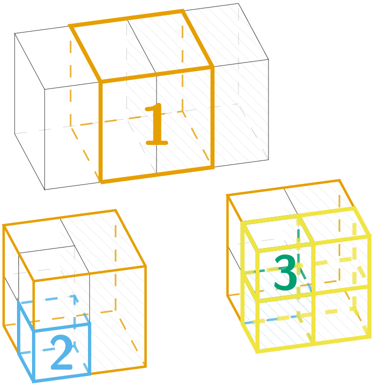
1. Probe center
2. Probe remaining 'quadrant'
   a. If success. . . 2-domino!
   b. Otherwise, more probes. . .



Rectilinear Strategies: 3D Domino – cont'd

- Idea: Intermediate configuration
- 3D 2-domino: $2n \times n \times n$

1. Probe center
2. Probe remaining 'quadrant'
   a. If success... 2-domino!
   b. Otherwise, more probes... but 4-domino guaranteed!

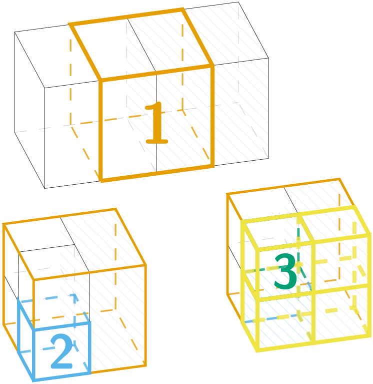


Rectilinear Strategies: 3D Domino – cont'd

- Idea: Intermediate configuration
- 3D 2-domino: $2n \times n \times n$

1. Probe center
2. Probe remaining 'quadrant'
   a. If success. . . 2-domino!
   b. Otherwise, more probes. . . but 4-domino guaranteed!

- Octants $\rightarrow$ 2-domino $\rightarrow$ 4-domino

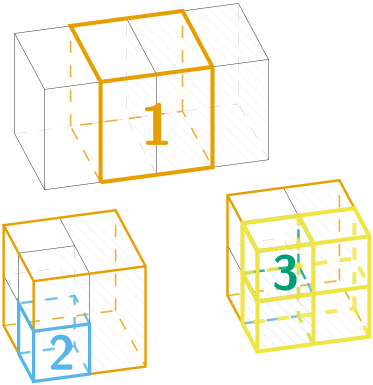


Rectilinear Strategies: 3D Domino – cont'd

- Idea: Intermediate configuration
- 3D 2-domino: $2n \times n \times n$

1. Probe center
2. Probe remaining 'quadrant'
   a. If success… 2-domino!
   b. Otherwise, more probes… but 4-domino guaranteed!

- Octants → 2-domino → 4-domino
- $P(n) \leq 3\lceil \log n \rceil + 4$

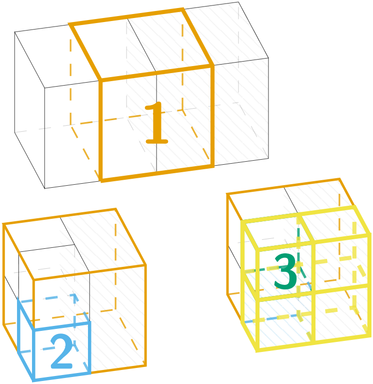


Rectilinear Strategies: 3D Domino – cont'd

- Idea: Intermediate configuration
- 3D 2-domino: $2n \times n \times n$

1. Probe center
2. Probe remaining 'quadrant'
   a. If success... 2-domino!
   b. Otherwise, more probes... but 4-domino guaranteed!

- Octants $\rightarrow$ 2-domino $\rightarrow$ 4-domino
- $P(n) \leq 3\lceil \log n \rceil + 4$
- Great but...

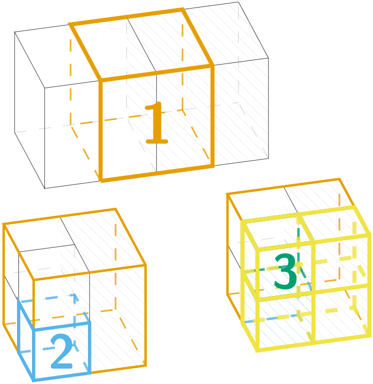


Rectilinear Strategies: 3D Domino – cont'd

- Idea: Intermediate configuration
- 3D 2-domino: $2n \times n \times n$

1. Probe center
2. Probe remaining 'quadrant'
   a. If success... 2-domino!
   b. Otherwise, more probes... but 4-domino guaranteed!

- Octants → 2-domino → 4-domino
- $P(n) \leq 3\lceil \log n \rceil + 4$
- Great but...
- Doesn't generalize further ☹

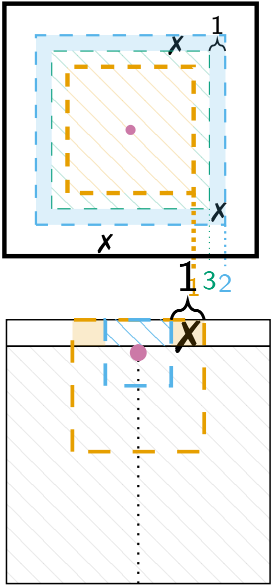


Rectilinear Strategies: Generalized CBS

- Good general algorithm?

Rectilinear Strategies: Generalized CBS

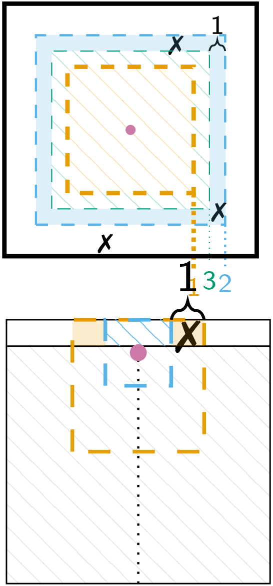
- Good general algorithm?
- Recall CBS—



Rectilinear Strategies: Generalized CBS

- Good general algorithm?

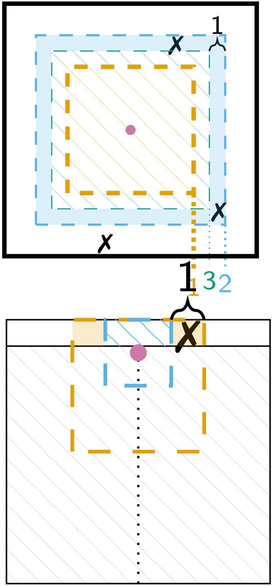
- Recall CBS—
- Binary search per dimension...



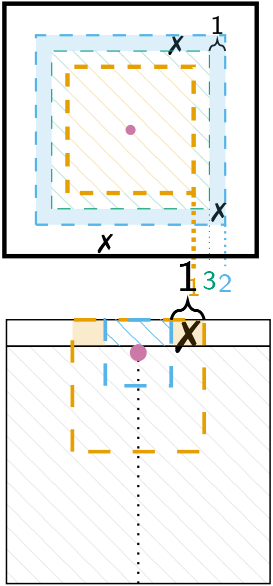
Rectilinear Strategies: Generalized CBS

- Good general algorithm?

- Recall CBS—
- Binary search per dimension...
- Generalize to kD?



- Good general algorithm?
- Recall CBS—
- Binary search per dimension...
- Generalize to kD? yes!

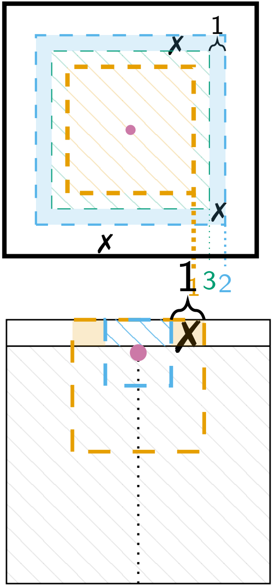


Rectilinear Strategies: Generalized CBS

- Good general algorithm?

- Recall CBS—
- Binary search per dimension. . .
- Generalize to kD? yes!

- Generalized CBS
- $P(n) \leq k\lceil\log n\rceil + g(k)$

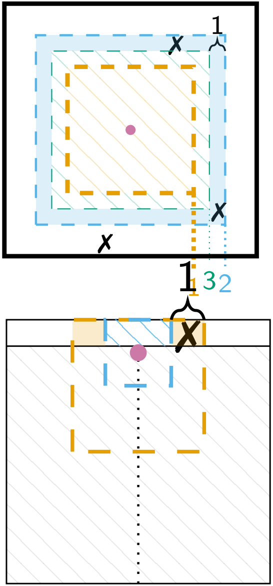


Rectilinear Strategies: Generalized CBS

- Good general algorithm?

- Recall CBS—
- Binary search per dimension...
- Generalize to kD? yes!

- Generalized CBS
- $P(n) \leq k\lceil \log n \rceil + g(k)$
- $D(n) \leq k\delta_{\min} + g(k)$



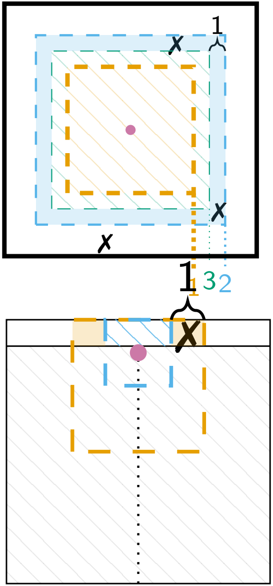
Rectilinear Strategies: Generalized CBS

- Good general algorithm?

- Recall CBS—
- Binary search per dimension...
- Generalize to kD? yes!

- Generalized CBS
- $P(n) \leq k\lceil \log n \rceil + g(k)$
- $D(n) \leq k\delta_{\min} + g(k)$

- How?



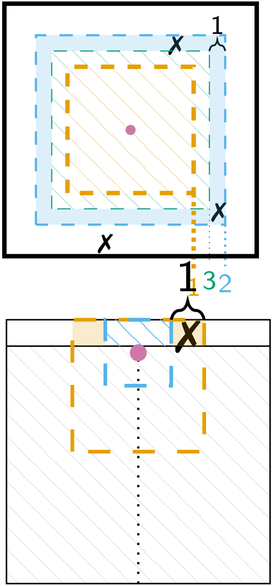
Rectilinear Strategies: Generalized CBS

- Good general algorithm?

- Recall CBS—
- Binary search per dimension...
- Generalize to kD? yes!

- Generalized CBS
- $P(n) \leq k\lceil \log n \rceil + g(k)$
- $D(n) \leq k\delta_{\min} + g(k)$

- How?
- Read the full paper :P



Rectilinear Strategies: Comparing # Probes

	Orthant Algorithm				Generalized CBS Algorithm			
k	σ	Avg	Max	Bound	σ	Avg	Max	Bound
1D	**0.00**	**0.95**	**0.95**	**1.00**	0.00	**0.95**	1.00	**1.00**
2D	0.18	2.14	2.85	3.00	**0.03**	1.93	2.00	2.15/2.25
3D	0.46	4.16	6.35	7.00	**0.07**	2.96	3.10	3.40/4.10
4D	0.98	8.02	13.2	15.0	**0.10**	**4.00**	**4.25**	**4.75/7.75**
5D	2.00	15.6	26.1	31.0	**0.14**	**5.06**	**5.40**	**6.20/16.8**
6D	4.02	30.9	50.6	63.0	**0.17**	**6.15**	**6.65**	**7.75/42.0**
7D	8.05	61.3	103	127	**0.21**	**7.26**	**7.95**	**9.40/116**
8D	16.1	122	209	**255**	**0.25**	**8.40**	**9.30**	11.2/336

	Domino Algorithms			
k	σ	Avg	Max	Bound
2D	0.04	**1.92**	**1.95**	**2.05**
3D	0.11	**2.92**	**3.05**	**3.20**

Rectilinear Strategies: Comparing Distance Traveled

	Orthant Algorithm				Generalized CBS Algorithm			
k	σ	Avg	Max	Bound	σ	Avg	Max	Bound
1D	$\sim 10^4$	27.8	$\sim 10^7$	$\sim 10^6$	**0.00**	**1.00**	**1.00**	**1.00**
2D	17.1	8.00	$\sim 10^4$	$\sim 10^6$	**0.29**	**1.50**	**2.00**	**2.00**
3D	8.14	12.0	$\sim 10^3$	$\sim 10^7$	**0.41**	**2.00**	**3.00**	**3.00**
4D	10.4	21.3	$\sim 10^3$	$\sim 10^7$	**0.50**	**2.50**	**3.99**	**4.00**
5D	16.7	40.0	$\sim 10^3$	$\sim 10^7$	**0.58**	**3.00**	**4.97**	**5.00**
6D	29.7	76.8	$\sim 10^3$	$\sim 10^8$	**0.65**	**3.50**	**5.94**	**6.00**
7D	55.2	149	$\sim 10^3$	$\sim 10^8$	**0.71**	**4.00**	**6.86**	**7.00**
8D	105	293	$\sim 10^3$	$\sim 10^8$	**0.76**	**4.50**	**7.76**	**8.00**

	Domino Algorithms			
k	σ	Avg	Max	Bound
2D	17.2	7.68	$\sim 10^4$	$\sim 10^6$
3D	7.17	11.0	$\sim 10^3$	$\sim 10^7$

Rectilinear Strategies: Comparing # Responses

	Orthant Algorithm				Generalized CBS Algorithm			
k	σ	Avg	Max	Bound	σ	Avg	Max	Bound
1D	0.11	0.47	**0.95**	**1.00**	**0.11**	**0.47**	**0.95**	1.05
2D	**0.09**	**0.71**	**0.95**	**1.00**	0.15	0.94	1.75	2.10
3D	**0.07**	**0.83**	**0.95**	**1.00**	0.18	1.42	2.45	3.15
4D	**0.05**	**0.89**	**0.95**	**1.00**	0.21	1.88	3.00	4.20
5D	**0.04**	**0.92**	**0.95**	**1.00**	0.23	2.33	3.50	5.25
6D	**0.03**	**0.94**	**0.95**	**1.00**	0.25	2.78	4.20	6.30
7D	**0.02**	**0.94**	**0.95**	**1.00**	0.27	3.22	4.65	7.35
8D	**0.01**	**0.95**	**0.95**	**1.00**	0.28	3.65	5.25	8.40

	Domino Algorithms			
k	σ	Avg	Max	Bound
2D	0.15	0.93	1.70	2.05
3D	0.18	1.39	2.35	3.20

Open Problems

- 2D rectilinear algs. work under both L_1 and L_∞ norms
  Confession: Only L_∞ works for higher dimensions
  Can we adapt general algs. for L_1?

- $g(k)$ constant in generalized CBS algorithm pretty large (3^k)
  Under reasonable assumptions can reduce to k^2
  Can we do better?

- Distance traveled of $\delta_{\min}$ instead of $k\delta_{\min}$?

- Mix & match distance metrics

- Other norms?