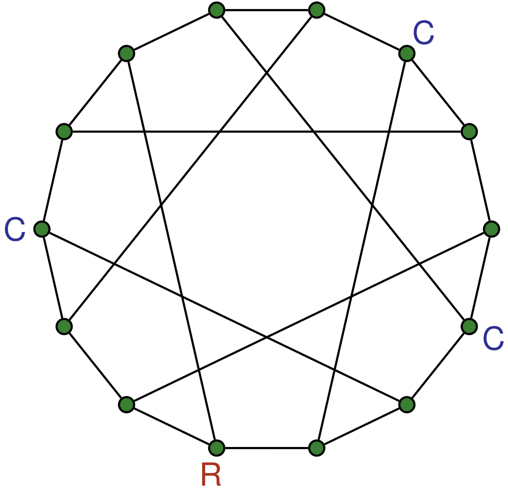
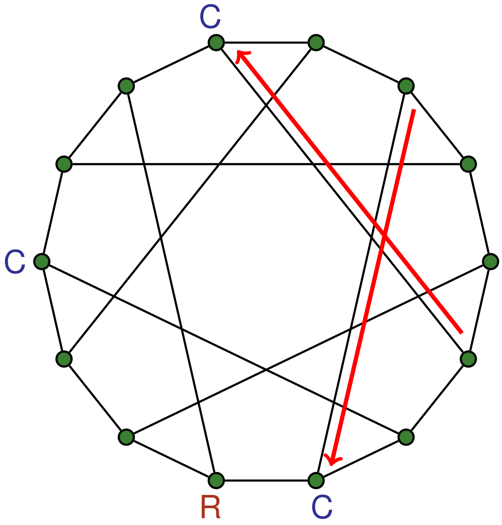
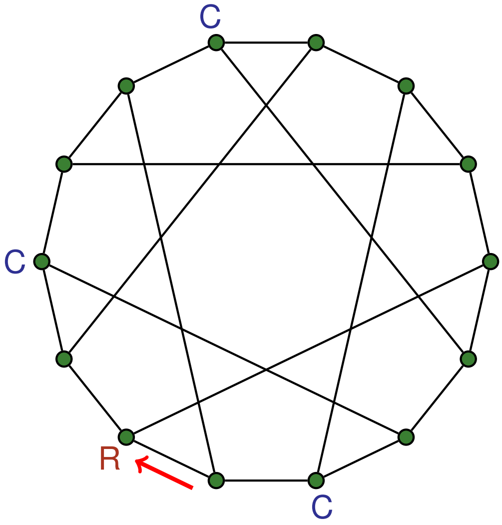
How long does it take to catch a robber?

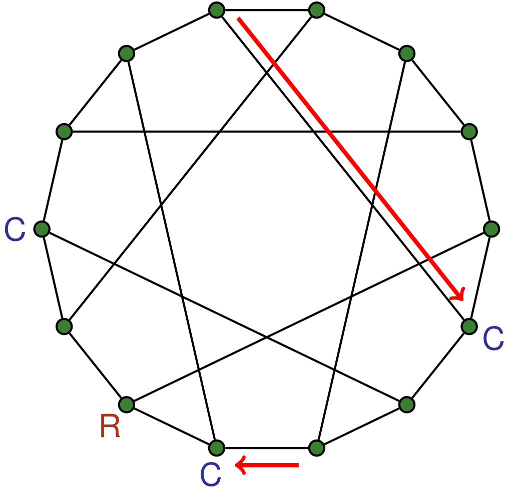
Bill Kinnersley

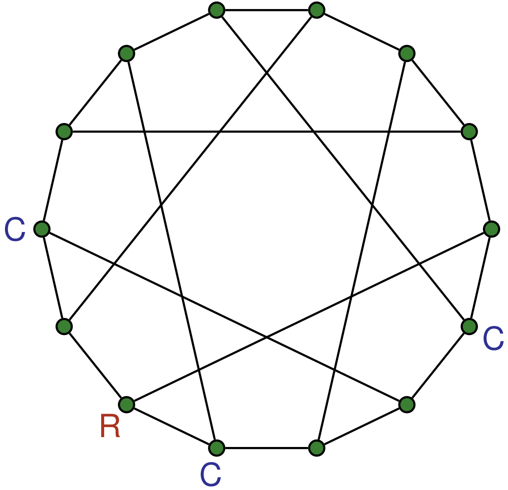
Department of Mathematics
Ryerson University
wkinners@ryerson.ca

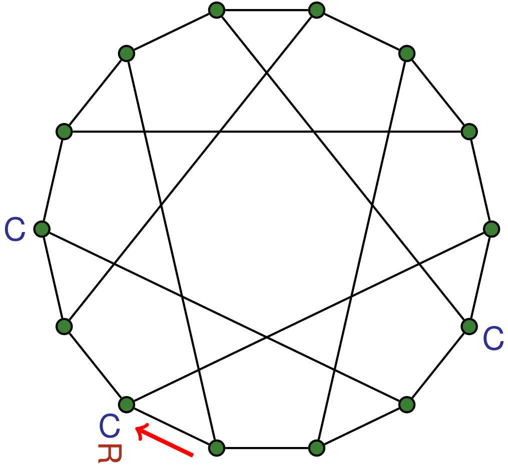

C
C
C
R


C
C
R
C


C
C
R
C


C
C
R
C


C
C
R
C


C
C
C
R

Cops and Robbers

Cops and Robbers: the mother of all graph games.

Cops and Robbers

Cops and Robbers: the mother of all graph games.

- Two teams: one robber versus one or more cops.
- Perfect information: everyone always knows everything.
- Players occupy vertices of some graph.
- Cops choose initial positions first, followed by the robber.
- Players alternate turns: cops first, then robber. On each turn, players may either move to neighboring vertices, or stand still.
- If the robber ever occupies the same vertex as some cop, the cops win. The robber wins by evading capture forever.

Cops and Robbers

Cops and Robbers: the mother of all graph games.

- Two teams: one robber versus one or more cops.
- Perfect information: everyone always knows everything.
- Players occupy vertices of some graph.
- Cops choose initial positions first, followed by the robber.
- Players alternate turns: cops first, then robber. On each turn, players may either move to neighboring vertices, or stand still.
- If the robber ever occupies the same vertex as some cop, the cops win. The robber wins by evading capture forever.

A natural question: how many cops are needed to force a win?

Cops and Robbers

Cops and Robbers: the mother of all graph games.

- Two teams: one robber versus one or more cops.
- Perfect information: everyone always knows everything.
- Players occupy vertices of some graph.
- Cops choose initial positions first, followed by the robber.
- Players alternate turns: cops first, then robber. On each turn, players may either move to neighboring vertices, or stand still.
- If the robber ever occupies the same vertex as some cop, the cops win. The robber wins by evading capture forever.

A natural question: how many cops are needed to force a win?

Definition

Given a graph G, the minimum number of cops needed to capture a robber on G is the cop number of G, denoted $c(G)$.

(Note: $|V(G)|$ cops always suffice.)

Capture time

A different spin on the problem: if we play with $c(G)$ cops, how long can the robber evade capture?

Capture time

A different spin on the problem: if we play with $c(G)$ cops, how long can the robber evade capture?

Definition

The capture time of G is the length of the game on G, under optimal play, with $c(G)$ cops.
Notation: $\mathrm{capt}(G)$

Capture time

A different spin on the problem: if we play with $c(G)$ cops, how long can the robber evade capture?

Definition

The capture time of G is the length of the game on G, under optimal play, with $c(G)$ cops.
Notation: $\mathrm{capt}(G)$

Theorem (Bonato, Golovach, Hahn, Kratochvíl ’09 and Gavenčiak ’11)

Let G be an n-vertex graph (with $n \geq 7$). If $c(G) = 1$, then $\mathrm{capt}(G) \leq n - 4$, and this is tight (even for planar graphs).

Capture time

A different spin on the problem: if we play with $c(G)$ cops, how long can the robber evade capture?

Definition

The capture time of G is the length of the game on G, under optimal play, with $c(G)$ cops.
Notation: $\mathrm{capt}(G)$

Theorem (Bonato, Golovach, Hahn, Kratochvíl '09 and Gavenčiak '11)

Let G be an n-vertex graph (with $n \geq 7$). If $c(G) = 1$, then $\mathrm{capt}(G) \leq n - 4$, and this is tight (even for planar graphs).

Best general bound: $\mathrm{capt}(G) \leq n\binom{n+c(G)-2}{c(G)} + 1$. (Not a very good bound!)

A polynomial bound on $\mathrm{capt}(G)$ (not depending on $c(G)$) would imply that computing $c(G)$ is PSPACE-complete.

Capture time

Theorem (Mehrabian '10)

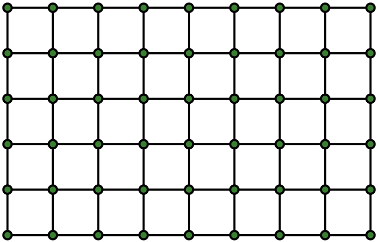
If G is the Cartesian product of two trees, then $\mathrm{capt}(G) = \lfloor \mathrm{diam}(G)/2 \rfloor$.

Capture time

Theorem (Mehrabian '10)

If G is the Cartesian product of two trees, then $\mathrm{capt}(G) = \lfloor \mathrm{diam}(G)/2 \rfloor$.

In particular, the capture time of the $m \times n$ grid is $\left\lfloor \frac{m+n}{2} \right\rfloor - 1$.

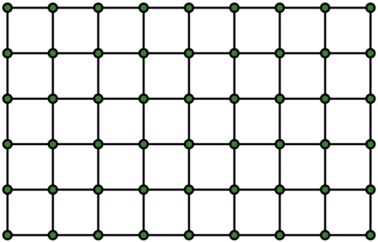


Capture time

Theorem (Mehrabian '10)

If G is the Cartesian product of two trees, then $\mathrm{capt}(G) = \lfloor \mathrm{diam}(G)/2 \rfloor$.

In particular, the capture time of the $m \times n$ grid is $\left\lfloor \frac{m+n}{2} \right\rfloor - 1$.



Natural next step: higher-dimensional grids.

Hypercubes

Our focus: the n-dimensional hypercube, Q_n.

- 2^n vertices – one for each n-bit binary string.
- Two vertices are joined by an edge if they differ in exactly one position.

Hypercubes

Our focus: the n-dimensional hypercube, Q_n.

- 2^n vertices – one for each n-bit binary string.
- Two vertices are joined by an edge if they differ in exactly one position.

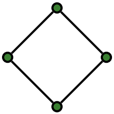
Q_1

Hypercubes

Our focus: the n-dimensional hypercube, Q_n.

- 2^n vertices – one for each n-bit binary string.
- Two vertices are joined by an edge if they differ in exactly one position.

Q_1



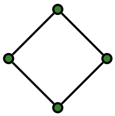
Q_2

Hypercubes

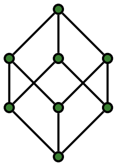
Our focus: the n-dimensional hypercube, Q_n.

- 2^n vertices – one for each n-bit binary string.
- Two vertices are joined by an edge if they differ in exactly one position.

Q_1



Q_2



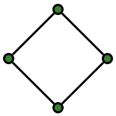
Q_3

Hypercubes

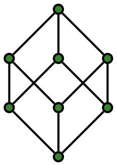
Our focus: the n-dimensional hypercube, Q_n.

- 2^n vertices – one for each n-bit binary string.
- Two vertices are joined by an edge if they differ in exactly one position.

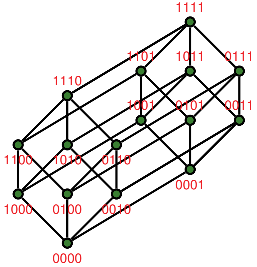
Q_1



Q_2



Q_3

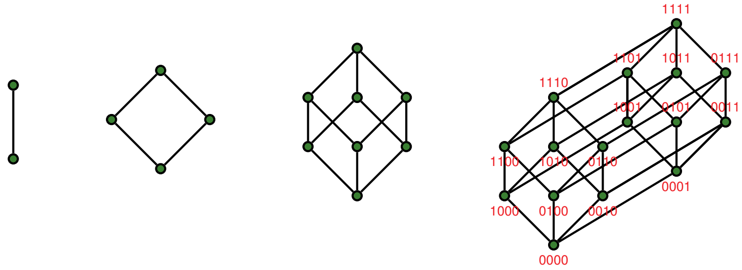


Q_4

Hypercubes

Our focus: the n-dimensional hypercube, Q_n.

- 2^n vertices – one for each n-bit binary string.
- Two vertices are joined by an edge if they differ in exactly one position.



Q_1 Q_2 Q_3 Q_4

Theorem (Bonato, Gordinowicz, Kinnersley, Prałat '13+)

The n-dimensional hypercube has capture time $\Theta(n \ln n)$.

Capture time – upper bound

Upper bound: $\mathrm{capt}(Q_n) \leq (1 + o(1))n \log_2 n$.

We need a strategy for the cops.

Capture time – upper bound

Upper bound: $\mathrm{capt}(Q_n) \leq (1 + o(1))n \log_2 n$.

We need a strategy for the cops.

Maamoun, Meyniel '87: $c(Q_n) = \lceil \frac{n+1}{2} \rceil$

How can $\lceil \frac{n+1}{2} \rceil$ cops catch the robber? Their idea:

Capture time – upper bound

Upper bound: $\mathrm{capt}(Q_n) \le (1 + o(1))n \log_2 n$.

We need a strategy for the cops.

Maamoun, Meyniel '87: $c(Q_n) = \lceil \frac{n+1}{2} \rceil$

How can $\lceil \frac{n+1}{2} \rceil$ cops catch the robber? Their idea:

- View vertices of Q_n as binary n-tuples.

Capture time – upper bound

Upper bound: $\mathrm{capt}(Q_n) \le (1+o(1))n\log_2 n$.

We need a strategy for the cops.

Maamoun, Meyniel '87: $c(Q_n) = \lceil \frac{n+1}{2} \rceil$

How can $\lceil \frac{n+1}{2} \rceil$ cops catch the robber? Their idea:

- View vertices of Q_n as binary n-tuples.
- "Assign" two coordinates to each cop (cop i gets coords $2i-1, 2i$).

Capture time – upper bound

Upper bound: $\mathrm{capt}(Q_n) \leq (1 + o(1))n \log_2 n$.

We need a strategy for the cops.

Maamoun, Meyniel '87: $c(Q_n) = \lceil \frac{n+1}{2} \rceil$

How can $\lceil \frac{n+1}{2} \rceil$ cops catch the robber? Their idea:

- View vertices of Q_n as binary n-tuples.
- "Assign" two coordinates to each cop (cop i gets coords $2i-1, 2i$).
- Each cop ignores his assigned coordinates, and tries to agree with the robber in all other coords.
  (capturing the "shadow" of the robber)

Capture time – upper bound

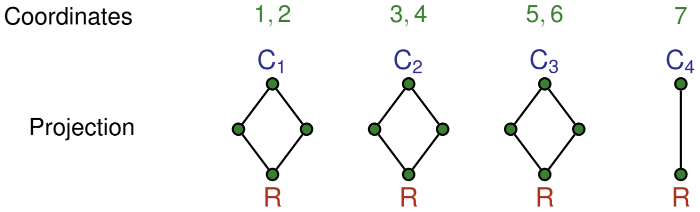
Upper bound: $\mathrm{capt}(Q_n) \leq (1 + o(1))n \log_2 n$.

We need a strategy for the cops.

Maamoun, Meyniel '87: $c(Q_n) = \lceil \frac{n+1}{2} \rceil$

How can $\lceil \frac{n+1}{2} \rceil$ cops catch the robber? Their idea:

- View vertices of Q_n as binary n-tuples.
- "Assign" two coordinates to each cop (cop i gets coords $2i - 1, 2i$).
- Each cop ignores his assigned coordinates, and tries to agree with the robber in all other coords.
  (capturing the "shadow" of the robber)



Capture time – upper bound

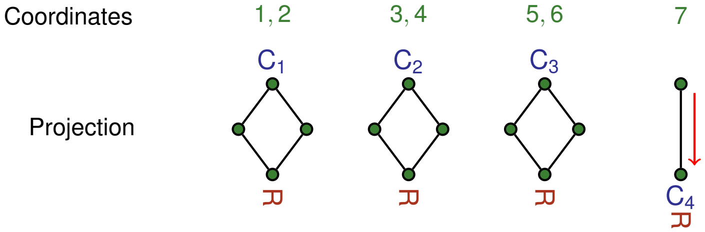
Upper bound: $\mathrm{capt}(Q_n) \le (1 + o(1))n \log_2 n$.

We need a strategy for the cops.

Maamoun, Meyniel '87: $c(Q_n) = \lceil \frac{n+1}{2} \rceil$

How can $\lceil \frac{n+1}{2} \rceil$ cops catch the robber? Their idea:

- View vertices of Q_n as binary n-tuples.
- "Assign" two coordinates to each cop (cop i gets coords $2i-1, 2i$).
- Each cop ignores his assigned coordinates, and tries to agree with the robber in all other coords.
  (capturing the "shadow" of the robber)



Capture time – upper bound

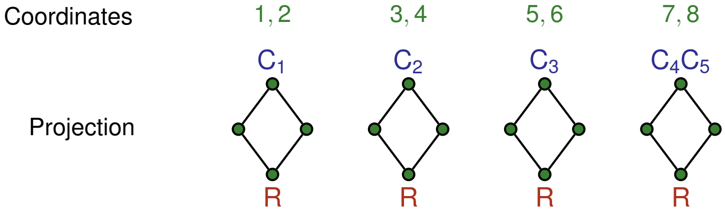
Upper bound: $\text{capt}(Q_n) \leq (1 + o(1))n \log_2 n$.

We need a strategy for the cops.

Maamoun, Meyniel '87: $c(Q_n) = \lceil \frac{n+1}{2} \rceil$

How can $\lceil \frac{n+1}{2} \rceil$ cops catch the robber? Their idea:

- View vertices of Q_n as binary n-tuples.
- "Assign" two coordinates to each cop (cop i gets coords $2i - 1, 2i$).
- Each cop ignores his assigned coordinates, and tries to agree with the robber in all other coords.
  (capturing the "shadow" of the robber)



Capture time – upper bound

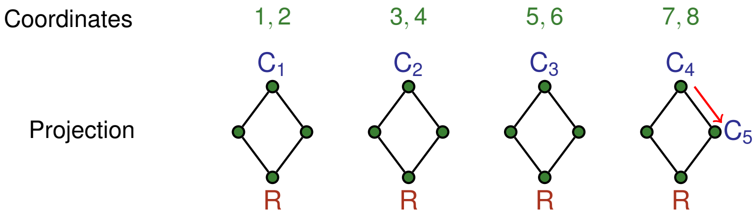
Upper bound: $\mathrm{capt}(Q_n) \leq (1 + o(1))n \log_2 n$.

We need a strategy for the cops.

Maamoun, Meyniel '87: $c(Q_n) = \lceil \frac{n+1}{2} \rceil$

How can $\lceil \frac{n+1}{2} \rceil$ cops catch the robber? Their idea:

- View vertices of Q_n as binary n-tuples.
- "Assign" two coordinates to each cop (cop i gets coords $2i-1, 2i$).
- Each cop ignores his assigned coordinates, and tries to agree with the robber in all other coords.
  (capturing the "shadow" of the robber)



Capture time – upper bound

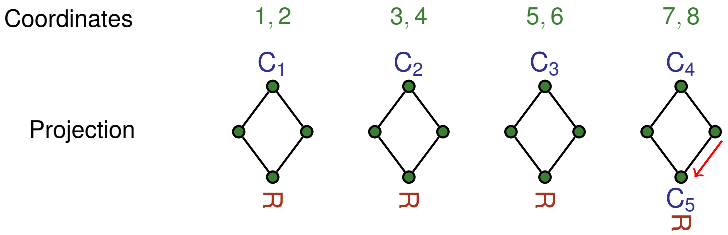
Upper bound: $\mathrm{capt}(Q_n) \leq (1 + o(1))n \log_2 n$.

We need a strategy for the cops.

Maamoun, Meyniel '87: $c(Q_n) = \lceil \frac{n+1}{2} \rceil$

How can $\lceil \frac{n+1}{2} \rceil$ cops catch the robber? Their idea:

- View vertices of Q_n as binary n-tuples.
- "Assign" two coordinates to each cop (cop i gets coords $2i - 1, 2i$).
- Each cop ignores his assigned coordinates, and tries to agree with the robber in all other coords.
  (capturing the "shadow" of the robber)



Capture time – upper bound

How do we reach this point, and how quickly can we do it?

Capture time – upper bound

How do we reach this point, and how quickly can we do it?

For simplicity, assume $n = 2^k$.

- All cops start on vertex $00\ldots0$.

Capture time – upper bound

How do we reach this point, and how quickly can we do it?

For simplicity, assume $n = 2^k$.

- All cops start on vertex $00\ldots0$.
- Partition cops into squads.
  Each squad has multiple cops, all of whom move together as one.

Capture time – upper bound

How do we reach this point, and how quickly can we do it?

For simplicity, assume $n = 2^k$.

- All cops start on vertex $00\ldots0$.
- Partition cops into squads.
  Each squad has multiple cops, all of whom move together as one.
- Each squad marks some coordinates active and the others inactive.
  Goal: capture the shadow of the robber in the active coords.

Capture time – upper bound

Initially, only two squads.

- The first sets coords $1, 2, \ldots, n/2$ active and the rest inactive.
- The second does the opposite: coords $n/2+1, n/2+2, \ldots, n$ active, rest inactive.

Capture time – upper bound

Initially, only two squads.

- The first sets coords $1, 2, \ldots, n/2$ active and the rest inactive.
- The second does the opposite: coords $n/2+1, n/2+2, \ldots, n$ active, rest inactive.

In each round, at least one squad closes in on the robber.

Squad A
01101000

Squad B
01011101

R
11100111

Capture time – upper bound

Initially, only two squads.

- The first sets coords $1, 2, \ldots, n/2$ active and the rest inactive.
- The second does the opposite: coords $n/2+1, n/2+2, \ldots, n$ active, rest inactive.

In each round, at least one squad closes in on the robber.

Squad A
01101000

Squad B
01011101

R
10100111

Capture time – upper bound

Initially, only two squads.

- The first sets coords $1, 2, \ldots, n/2$ active and the rest inactive.
- The second does the opposite: coords $n/2 + 1, n/2 + 2, \ldots, n$ active, rest inactive.

In each round, at least one squad closes in on the robber.

Squad A
00101000

Squad B
01011101

R
10100111

Capture time – upper bound

Initially, only two squads.

- The first sets coords $1, 2, \ldots, n/2$ active and the rest inactive.
- The second does the opposite: coords $n/2+1, n/2+2, \ldots, n$ active, rest inactive.

In each round, at least one squad closes in on the robber.

Squad A
00101000

Squad B
01010101

R
10100111

Capture time – upper bound

Initially, only two squads.

- The first sets coords $1, 2, \ldots, n/2$ active and the rest inactive.
- The second does the opposite: coords $n/2+1, n/2+2, \ldots, n$ active, rest inactive.

In each round, at least one squad closes in on the robber.

Squad A
00101000

Squad B
01010101

R
10100111

Once a squad achieves its goal, it splits in half.
Each “sub-squad” activates half their inactive coords.

Capture time – upper bound

Initially, only two squads.

- The first sets coords $1, 2, \ldots, n/2$ active and the rest inactive.
- The second does the opposite: coords $n/2+1, n/2+2, \ldots, n$ active, rest inactive.

In each round, at least one squad closes in on the robber.

Squad A
00101000

Squad B
01010101

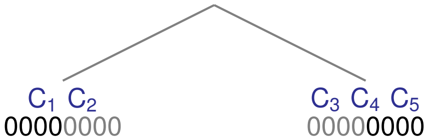
R
10100111

Once a squad achieves its goal, it splits in half.
Each "sub-squad" activates half their inactive coords.

Eventually, each squad has just one cop – with two inactive coords.

Capture time – upper bound

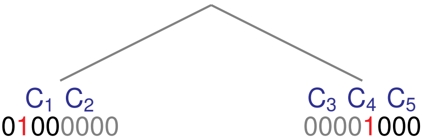
Example (Q_8)



Robber: 01110010

Capture time – upper bound

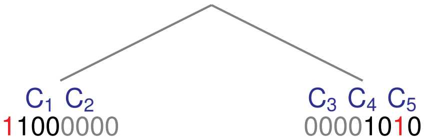
Example (Q_8)



Robber: 01111010

Capture time – upper bound

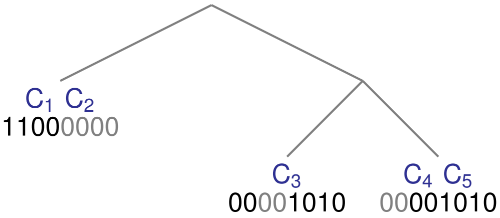
Example (Q_8)



Robber: 11111010

Capture time – upper bound

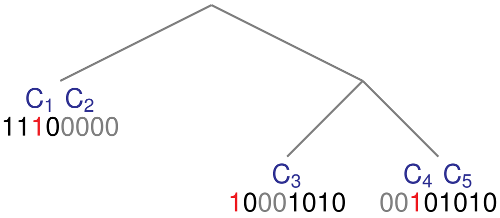
Example (Q_8)



Robber: 11111010

Capture time – upper bound

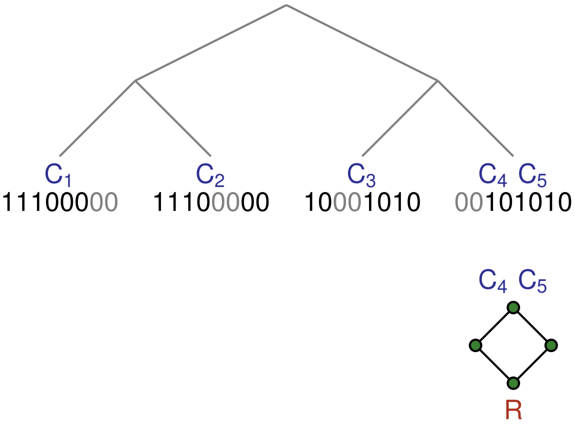
Example (Q_8)



Robber: 11101010

Capture time – upper bound

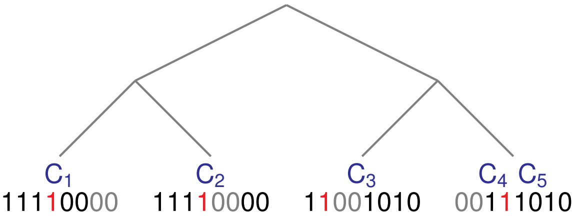
Example (Q_8)

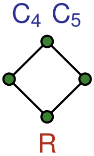


Robber: 11101010

Capture time – upper bound

Example (Q_8)

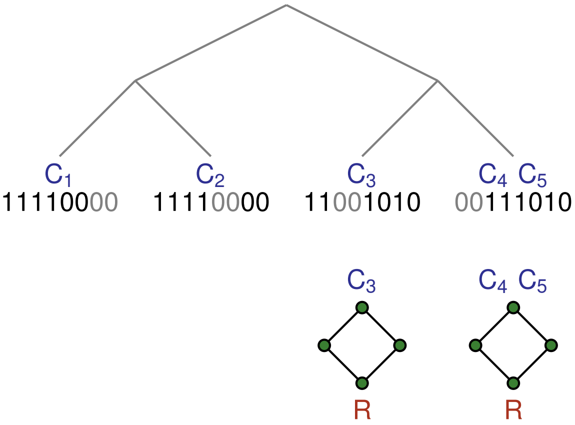




Robber: 11111010

Capture time – upper bound

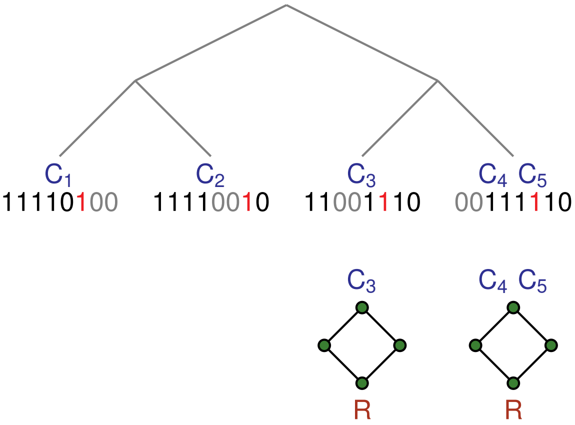
Example (Q_8)



Robber: 11111010

Capture time – upper bound

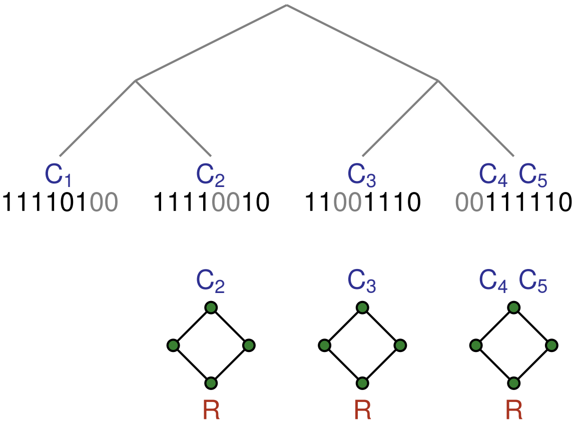
Example (Q_8)



Robber: 11111110

Capture time – upper bound

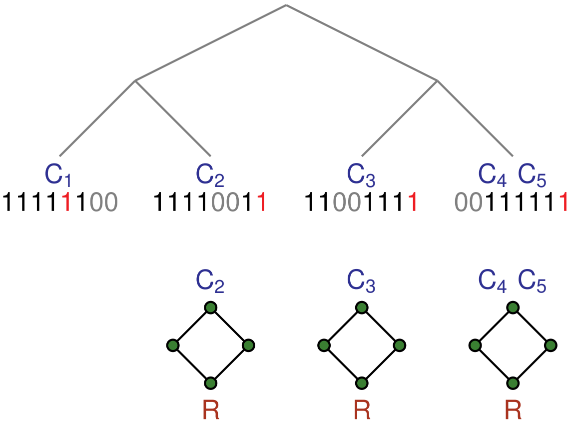
Example (Q_8)



Robber: 11111110

Capture time – upper bound

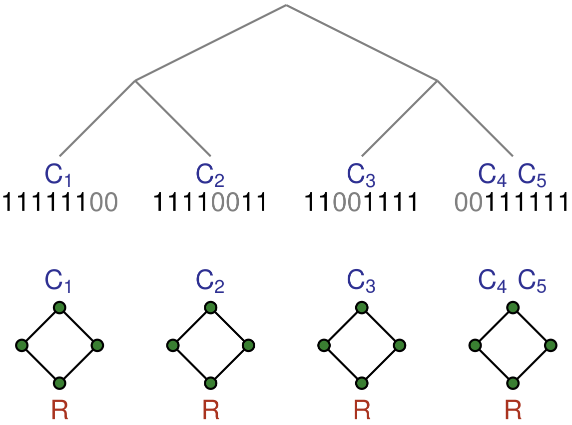
Example (Q_8)



Robber: 11111111

Capture time – upper bound

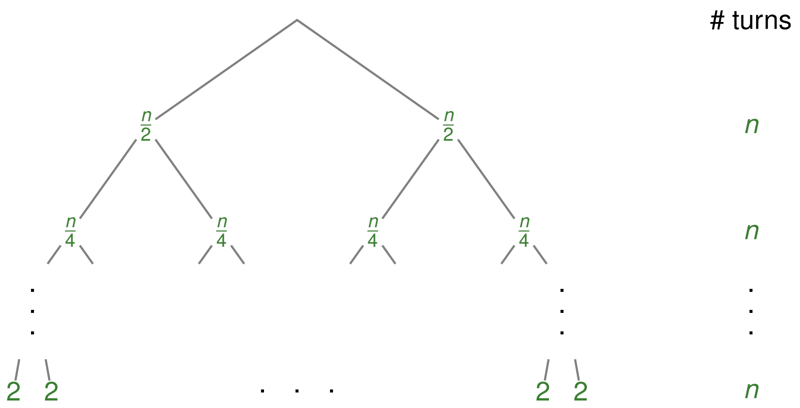
Example (Q_8)



Robber: 11111111

Capture time – upper bound

How long could this take?



Capture time – upper bound

When all is said and done, we have

$$\mathrm{capt}(Q_n) \leq n \lceil \log_2 n \rceil - \left\lfloor \frac{n-1}{2} \right\rfloor + 1.$$

Capture time – upper bound

When all is said and done, we have

$$\mathrm{capt}(Q_n) \leq n \lceil \log_2 n \rceil - \left\lfloor \frac{n-1}{2} \right\rfloor + 1.$$

In fact, we can say a bit more:

Theorem (Bonato, Gordinowicz, Kinnersley, Prałat 13+)

For every integer n, and all trees $T_1, T_2 \ldots, T_n$, we have

$$\mathrm{capt}(T_1 \,\Box\, T_2 \,\Box\, \cdots \,\Box\, T_n) \leq \left(\sum_{i=0}^{n-1} \mathrm{rad}(T_i) \right) \lceil \log_2 n \rceil - \left\lfloor \frac{n-1}{2} \right\rfloor + 1.$$

Capture time – lower bound

Lower bound: $\mathrm{capt}(Q_n) \geq (1 - o(1))\frac{1}{2} n \ln n$.

Strategy for the robber :

Capture time – lower bound

Lower bound: $\mathrm{capt}(Q_n) \geq (1 - o(1))\frac{1}{2} n \ln n$.

Strategy for the robber :

- Choose a starting location far from all cops (at least distance $n/4$).

Capture time – lower bound

Lower bound: $\mathrm{capt}(Q_n) \geq (1 - o(1))\frac{1}{2} n \ln n$.

Strategy for the robber :

- Choose a starting location far from all cops (at least distance $n/4$).
- Move randomly.

Capture time – lower bound

Lower bound: $\mathrm{capt}(Q_n) \geq (1 - o(1))\frac{1}{2} n \ln n$.

Strategy for the robber :

- Choose a starting location far from all cops (at least distance $n/4$).
- Move randomly.

How could this possibly be a good idea?

Capture time – lower bound

Lower bound: $\mathrm{capt}(Q_n) \geq (1 - o(1))\frac{1}{2} n \ln n$.

Strategy for the robber :

- Choose a starting location far from all cops (at least distance $n/4$).
- Move randomly.

How could this possibly be a good idea?

Analysis:

- Fix a strategy for the cops.
- We show that the probability of capture by one particular cop (within $\frac{1}{2} n \ln n$ rounds) is very small – so small that the probability of capture by any cop tends to 0.
- Consequently, some sequence of random choices for the robber "works" – so he has a deterministic strategy to survive.

Capture time – lower bound

Cops' optimal strategy: move greedily.
On each turn, move one step closer to the robber.

Capture time – lower bound

Cops' optimal strategy: move greedily.
On each turn, move one step closer to the robber.

Suppose cop, robber are at distance d. Consider one round of the game.

Capture time – lower bound

Cops' optimal strategy: move greedily.
On each turn, move one step closer to the robber.

Suppose cop, robber are at distance d. Consider one round of the game.

One possibility: robber moves away, cop moves closer. Still at distance d.
Probability of this: $(n-d)/n$.

Capture time – lower bound

Cops' optimal strategy: move greedily.
On each turn, move one step closer to the robber.

Suppose cop, robber are at distance d. Consider one round of the game.

One possibility: robber moves away, cop moves closer. Still at distance d.
Probability of this: $(n-d)/n$.

Else: robber approaches, cop moves even closer. Now at distance $d-2$.
Probability: d/n.

Capture time – lower bound

Cops' optimal strategy: move greedily.
On each turn, move one step closer to the robber.

Suppose cop, robber are at distance d. Consider one round of the game.

One possibility: robber moves away, cop moves closer. Still at distance d.
Probability of this: $(n - d)/n$.

Else: robber approaches, cop moves even closer. Now at distance $d - 2$.
Probability: d/n.

Cop decreases distance by 2 with probability d/n; otherwise, no change.

Capture time – lower bound

Cops' optimal strategy: move greedily.
On each turn, move one step closer to the robber.

Suppose cop, robber are at distance d. Consider one round of the game.

One possibility: robber moves away, cop moves closer. Still at distance d.
Probability of this: $(n - d)/n$.

Else: robber approaches, cop moves even closer. Now at distance $d - 2$.
Probability: d/n.

Cop decreases distance by 2 with probability d/n; otherwise, no change.

(Technicality: cop wants to stay at even distance. If at odd distance, sit still for one turn; after that, move greedily.)

Digression? Coupon collecting

Coupon-collector problem: a store produces m types of coupons. One coupon is distributed with each purchase (all types equally likely). A prudent(?) shopper wants to collect all m coupons. How many purchases must he make, on average?

Digression? Coupon collecting

Coupon-collector problem: a store produces m types of coupons. One coupon is distributed with each purchase (all types equally likely). A prudent(?) shopper wants to collect all m coupons. How many purchases must he make, on average?

With k coupons already collected, probability of seeing a new type is $(m-k)/m$. Expected number of trials needed:

$$\frac{m}{m} + \frac{m}{m-1} + \frac{m}{m-2} + \ldots + \frac{m}{1} = m \sum_{i=1}^{m} \frac{1}{i},$$

which tends to $m \ln m + O(1)$.

Digression? Coupon collecting

Coupon-collector problem: a store produces m types of coupons. One coupon is distributed with each purchase (all types equally likely). A prudent(?) shopper wants to collect all m coupons. How many purchases must he make, on average?

With k coupons already collected, probability of seeing a new type is $(m-k)/m$. Expected number of trials needed:

$$\frac{m}{m} + \frac{m}{m-1} + \frac{m}{m-2} + \ldots + \frac{m}{1} = m \sum_{i=1}^{m} \frac{1}{i},$$

which tends to $m \ln m + O(1)$.

So what?

Digression? Coupon collecting

Coupon-collector problem: a store produces m types of coupons. One coupon is distributed with each purchase (all types equally likely). A prudent(?) shopper wants to collect all m coupons. How many purchases must he make, on average?

With k coupons already collected, probability of seeing a new type is $(m-k)/m$. Expected number of trials needed:

$$\frac{m}{m} + \frac{m}{m-1} + \frac{m}{m-2} + \ldots + \frac{m}{1} = m \sum_{i=1}^{m} \frac{1}{i},$$

which tends to $m \ln m + O(1)$.

So what?

With d coupons uncollected, prob. of getting a new coupon: d/m

For cop/robber at distance d, prob. of getting closer: d/n

Capture time – lower bound

So, expected number of rounds needed to catch the robber equals expected length of coupon-collector process.

Capture time – lower bound

So, expected number of rounds needed to catch the robber equals expected length of coupon-collector process.

We need something stronger:

Lemma

Consider the coupon-collector process with m coupons in total, of which all but m_0 have already been collected. Let X be a random variable denoting the number of rounds needed to collect the remaining m_0 coupons. For $\varepsilon > 0$, we have

$$\mathbf{Pr}\left[X < (1-\varepsilon)(m-1)\ln m\right] \leq \exp(-m^{-1+\varepsilon} m_0).$$

i.e. we almost certainly need $(1 - o(1))m \ln m$ trials.

Capture time – lower bound

So, expected number of rounds needed to catch the robber equals expected length of coupon-collector process.

We need something stronger:

Lemma

Consider the coupon-collector process with m coupons in total, of which all but m_0 have already been collected. Let X be a random variable denoting the number of rounds needed to collect the remaining m_0 coupons. For $\varepsilon > 0$, we have

$$\mathbf{Pr}\left[X < (1-\varepsilon)(m-1)\ln m\right] \leq \exp(-m^{-1+\varepsilon} m_0).$$

i.e. we almost certainly need $(1-o(1))m \ln m$ trials.

Applying this to the game:

- Only $n/2$ "coupons" (since distance decreases by 2s)
- Initially, at least $n/8$ coupons uncollected (since robber starts at least distance $n/4$ away)

Capture time – lower bound

Lemma

Consider the coupon-collector process with m coupons in total, of which all but m_0 have already been collected. Let X be a random variable denoting the number of rounds needed to collect the remaining m_0 coupons. For $\varepsilon > 0$, we have

$$\mathbf{Pr}\left[X < (1-\varepsilon)(m-1)\ln m\right] \leq \exp(-m^{-1+\varepsilon} m_0).$$

Capture time – lower bound

Lemma

Consider the coupon-collector process with m coupons in total, of which all but m_0 have already been collected. Let X be a random variable denoting the number of rounds needed to collect the remaining m_0 coupons. For $\varepsilon > 0$, we have

$$\mathbf{Pr}\left[X < (1-\varepsilon)(m-1)\ln m\right] \le \exp(-m^{-1+\varepsilon} m_0).$$

Let $T = \frac{1}{2}(n-1)\ln n$ and $\varepsilon = \frac{\ln(5\ln n)}{\ln n} = o(1)$.

Probability that a given cop captures the robber in under T rounds is at most

$$\exp\left(-\left(\frac{n}{2}\right)^{-1+\varepsilon}\cdot\frac{n}{8}\right) = \exp\left(-\frac{(n/2)^{\varepsilon}}{4}\right).$$

Capture time – lower bound

Lemma

Consider the coupon-collector process with m coupons in total, of which all but m_0 have already been collected. Let X be a random variable denoting the number of rounds needed to collect the remaining m_0 coupons. For $\varepsilon > 0$, we have

$$\mathbf{Pr}\left[X < (1-\varepsilon)(m-1)\ln m\right] \leq \exp(-m^{-1+\varepsilon} m_0).$$

Let $T = \frac{1}{2}(n-1)\ln n$ and $\varepsilon = \frac{\ln(5\ln n)}{\ln n} = o(1)$.

Probability that a given cop captures the robber in under T rounds is at most

$$\exp\left(-\left(\frac{n}{2}\right)^{-1+\varepsilon} \cdot \frac{n}{8}\right) = \exp\left(-\frac{(n/2)^{\varepsilon}}{4}\right).$$

Probability that any cop captures the robber in under T rounds is at most

$$\left\lceil \frac{n+1}{2} \right\rceil \exp\left(-\frac{(n/2)^{\varepsilon}}{4}\right) = o(1).$$

Capture time – lower bound

Unlike the upper bound, this doesn't extend well to other graphs. However...

Capture time – lower bound

Unlike the upper bound, this doesn't extend well to other graphs. However...

For suitable choice of ε, we can make the argument work against any polynomial number of cops.

Capture time – lower bound

Unlike the upper bound, this doesn't extend well to other graphs. However...

For suitable choice of ε, we can make the argument work against any polynomial number of cops.

- How does the capture time change as we add more cops?

Capture time – lower bound

Unlike the upper bound, this doesn't extend well to other graphs. However...

For suitable choice of ε, we can make the argument work against any polynomial number of cops.

- How does the capture time change as we add more cops?
- Is the "drunk robber" still effective against many cops? (I think so.)

Capture time – lower bound

Unlike the upper bound, this doesn't extend well to other graphs. However...

For suitable choice of ε, we can make the argument work against any polynomial number of cops.

- How does the capture time change as we add more cops?
- Is the "drunk robber" still effective against many cops? (I think so.)
- Is the "drunk robber" still effective on larger grids? (Maybe not.)

Capture time of Q_n

The cop and robber strategies together yield $\mathrm{capt}(Q_n) = \Theta(n \ln n)$.

Capture time of Q_n

The cop and robber strategies together yield $\mathrm{capt}(Q_n) = \Theta(n \ln n)$.

But on closer inspection:

$$(1 - o(1))\frac{1}{2} n \ln n \leq \mathrm{capt}(Q_n) \leq (1 + o(1)) n \log_2 n.$$

Capture time of Q_n

The cop and robber strategies together yield $\mathrm{capt}(Q_n) = \Theta(n \ln n)$.

But on closer inspection:

$$(1 - o(1))\frac{1}{2} n \ln n \leq \mathrm{capt}(Q_n) \leq (1 + o(1)) n \log_2 n.$$

We conjecture that the lower bound is tight (i.e., drunkenness is optimal).

Other problems to consider:

Capture time of Q_n

The cop and robber strategies together yield $\mathrm{capt}(Q_n) = \Theta(n \ln n)$.

But on closer inspection:

$$(1 - o(1))\frac{1}{2} n \ln n \leq \mathrm{capt}(Q_n) \leq (1 + o(1)) n \log_2 n.$$

We conjecture that the lower bound is tight (i.e., drunkenness is optimal).

Other problems to consider:

- Compute $\mathrm{capt}(G)$ for other classes of graphs (larger grids, random graphs, ...)

Capture time of Q_n

The cop and robber strategies together yield $\mathrm{capt}(Q_n) = \Theta(n \ln n)$.

But on closer inspection:

$$(1 - o(1))\frac{1}{2} n \ln n \leq \mathrm{capt}(Q_n) \leq (1 + o(1)) n \log_2 n.$$

We conjecture that the lower bound is tight (i.e., drunkenness is optimal).

Other problems to consider:

- Compute $\mathrm{capt}(G)$ for other classes of graphs (larger grids, random graphs, ...)
- When is drunkenness optimal? Almost optimal? How bad can it be? Work has been done on the drunk robber vs. one cop. How do the results compare?

Capture time of Q_n

The cop and robber strategies together yield $\mathrm{capt}(Q_n) = \Theta(n \ln n)$.

But on closer inspection:

$$(1 - o(1))\frac{1}{2} n \ln n \leq \mathrm{capt}(Q_n) \leq (1 + o(1)) n \log_2 n.$$

We conjecture that the lower bound is tight (i.e., drunkenness is optimal).

Other problems to consider:

- Compute $\mathrm{capt}(G)$ for other classes of graphs (larger grids, random graphs, ...)
- When is drunkenness optimal? Almost optimal? How bad can it be? Work has been done on the drunk robber vs. one cop. How do the results compare?
- Is $\mathrm{capt}(G)$ always bounded by some polynomial in $|V(G)|$? (If yes, then computing the cop number is PSPACE-complete. If no, a constructive proof would still be interesting.)

Thanks

Thank you!