The Logic of Random Graphs

Peleg Michaeli

Tel-Aviv University

April 13, 2011

Lecture material by Joel Spencer

Motivation

Problems

- Large graphs are very hard to analyze.

Motivation

Problems

- Large graphs are very hard to analyze.
- Analyzing a large graph "manually" is rather primitive.

Motivation

Problems

- Large graphs are very hard to analyze.
- Analyzing a large graph "manually" is rather primitive.
- Large graphs are everywhere: social networks, actors playing together in movies, biology, etc.

Motivation

Problems

- Large graphs are very hard to analyze.
- Analyzing a large graph "manually" is rather primitive.
- Large graphs are everywhere: social networks, actors playing together in movies, biology, etc.

A Possible Solution

Analyzing the typical behaviour of a graph with "similar properties". There are several known models for doing this.

The Erdős—Rényi Models

Definition

$G \in G(n,p)$ if G has n vertices, and every edge is included in the graph with probability p.

The Erdős—Rényi Models

Definition

$G \in G(n, p)$ if G has n vertices, and every edge is included in the graph with probability p.

Definition

$G \in G(n, m)$ if G has n vertices and m edges, chosen uniformly from all $\binom{n}{2}$ possible edges.

The Erdős—Rényi Models

Definition

$G \in G(n, p)$ if G has n vertices, and every edge is included in the graph with probability p.

Definition

$G \in G(n, m)$ if G has n vertices and m edges, chosen uniformly from all $\binom{n}{2}$ possible edges.

Example

Let $G \in G(n, p)$. Then $\lim_{n\to\infty} \mathbb{P}(G \text{ has an edge}) = 1$.

Example Application

Theorem

Let R_k be the k'th diagonal Ramsey number. Then: $R_k > 2^{k/2}$.

Example Application

Theorem

Let R_k be the k'th diagonal Ramsey number. Then: $R_k > 2^{k/2}$.

Proof.

Let $G \in G(n, \frac{1}{2})$, $n \leq 2^{k/2}$. We calculate:

$$\mathbb{P}(\omega(G) \geq k) \leq \binom{n}{k} 2^{-\binom{k}{2}} \leq \left(\frac{n}{2 \cdot 2^{-(k-1)/2}}\right)^k \leq \left(\frac{\sqrt{2}}{2}\right)^k < \frac{1}{2}$$

□

The Graph Language

First-Order Logic: Our Language

- variables: $x_1, x_2, \ldots$

The Graph Language

First-Order Logic: Our Language

- variables: $x_1, x_2, \ldots$
- two binary relations: $=$ and $\sim$

The Graph Language

First-Order Logic: Our Language

- variables: $x_1, x_2, \ldots$
- two binary relations: $=$ and $\sim$
- quantifiers: $\forall$ and $\exists$

The Graph Language

First-Order Logic: Our Language

- variables: $x_1, x_2, \ldots$
- two binary relations: $=$ and $\sim$
- quantifiers: $\forall$ and $\exists$
- connectives: $\vee, \wedge, \rightarrow, \neg$

The Graph Language

First-Order Logic: Our Language

- variables: $x_1, x_2, \ldots$
- two binary relations: $=$ and $\sim$
- quantifiers: $\forall$ and $\exists$
- connectives: $\vee, \wedge, \rightarrow, \neg$

Note

We will assume the following two axioms:

The Graph Language

First-Order Logic: Our Language

- variables: $x_1, x_2, \ldots$
- two binary relations: $=$ and $\sim$
- quantifiers: $\forall$ and $\exists$
- connectives: $\vee, \wedge, \rightarrow, \neg$

Note

We will assume the following two axioms:

- $\forall x \; \neg x \sim x$

The Graph Language

First-Order Logic: Our Language

- variables: $x_1, x_2, \ldots$
- two binary relations: $=$ and $\sim$
- quantifiers: $\forall$ and $\exists$
- connectives: $\vee, \wedge, \rightarrow, \neg$

Note

We will assume the following two axioms:

- $\forall x \; \neg x \sim x$
- $\forall x \forall y \; x \sim y \rightarrow y \sim x$

Example

Example

Let $A =$ "$\exists x \exists y \exists z (x \sim y) \wedge (y \sim z) \wedge (z \sim x)$". In that case, $G \models A$ if and only if G (thought of as a graph) contains a triangle.

Example

Example

Let $A =$ "$\exists x \exists y \exists z (x \sim y) \wedge (y \sim z) \wedge (z \sim x)$". In that case, $G \models A$ if and only if G (thought of as a graph) contains a triangle.

Proposition

$\lim_{n\to\infty} \mathbb{P}\left(G(n, \frac{1}{2}) \models A\right) = 1.$

Example

Example

Let $A =$ "$\exists x \exists y \exists z (x \sim y) \wedge (y \sim z) \wedge (z \sim x)$". In that case, $G \models A$ if and only if G (thought of as a graph) contains a triangle.

Proposition

$\lim_{n \to \infty} \mathbb{P}\left(G(n, \frac{1}{2}) \models A\right) = 1.$

Proof.

Partition the vertices of G into sets of 3. Each set contains a triangle with probability $1/8$, hence the probability that G does not contain a triangle is no higher than $\left(\frac{7}{8}\right)^{n/3}$. □

Almost Sure Theories

Definition

An almost sure theory is the set of all sentences A holding almost surely (with respect to some $p = p(n)$).

Almost Sure Theories

Definition

An almost sure theory is the set of all sentences A holding almost surely (with respect to some $p = p(n)$).

Theorem

An almost sure theory is a theory.

Almost Sure Theories

Definition

An almost sure theory is the set of all sentences A holding almost surely (with respect to some $p = p(n)$).

Theorem

An almost sure theory is a theory.

Proof.

Suppose B is deduced from an almost sure theory T; hence, it can be deduced from a finite subset of T, $A_1, \ldots, A_k$.
$\mathbb{P}(G(n,p) \models \neg A_1 \wedge \ldots \wedge \neg A_k) \leq \sum_{i=1}^{k} \mathbb{P}(G(n,p) \models \neg A_i)$ □

Almost Sure Theories

Theorem

An almost sure theory T is consistent.

Almost Sure Theories

Theorem

An almost sure theory T is consistent.

Proof.

The sentence "False" does not hold almost surely, hence it is not in T. □

Completeness

Theorem

Let T be a theory with no finite models. Then T is complete iff all of its infinite models are elementarily equivalent.

Completeness

Proof.

- Suppose T is complete. Let A be a first-order sentence. Either $T \models A$ or $T \models \neg A$, hence all infinite models G of T satisfy either A or $\neg A$, respectively.

Completeness

Proof.

- Suppose T is complete. Let A be a first-order sentence. Either $T \models A$ or $T \models \neg A$, hence all infinite models G of T satisfy either A or $\neg A$, respectively.
- Suppose T is incomplete. Let B be a first-order sentence for which neither $T \models B$ nor $T \models \neg B$. Let T^+ be the theory given by adding B to T, and let T^- be the theory given by adding $\neg B$. Both are consistent, hence by Gödel both have models, G^+ and G^-, which are models of T, but are not elementarily equivalent, since they disagree on B.

□

The Zero-One Law

Definition

We say that $p = p(n)$ satisfies the Zero-One Law if for every first-order sentence A, the following holds:
$\lim_{n\to\infty} \mathbb{P}\left(G(n,p(n)) \models A\right) \in \{0,1\}$.

The Zero-One Law

Definition

We say that $p = p(n)$ satisfies the Zero-One Law if for every first-order sentence A, the following holds:
$\lim_{n\to\infty} \mathbb{P}\left(G(n, p(n)) \models A\right) \in \{0, 1\}$.

Theorem (Fagin)

The constant function $p(n) \equiv \frac{1}{2}$ satisfies the Zero-One Law.

The Zero-One Law

Definition

We say that $p = p(n)$ satisfies the Zero-One Law if for every first-order sentence A, the following holds:
$\lim_{n\to\infty} \mathbb{P}\left(G(n, p(n)) \models A\right) \in \{0, 1\}$.

Theorem (Fagin)

The constant function $p(n) \equiv \frac{1}{2}$ satisfies the Zero-One Law.
Note: This can be generalised to any constant $p(n) \equiv p$.

Alice's Restaurant Property

Definition

For any non-negative integers r, s, let $A_{r,s}$ be the following statement: "For any distinct $x_1, \ldots, x_r$ and $y_1, \ldots, y_s$ there exists a vertex z such that $z \sim x_i$ for all i and $\neg z \sim y_i$ for all i."

Alice's Restaurant Property

Definition

For any non-negative integers r, s, let $A_{r,s}$ be the following statement: "For any distinct $x_1, \ldots, x_r$ and $y_1, \ldots, y_s$ there exists a vertex z such that $z \sim x_i$ for all i and $\neg z \sim y_i$ for all i."
Note: This is a first-order sentence.

Alice's Restaurant Property

Proposition

$\forall r, s \geq 0$, $A_{r,s}$ *holds almost surely.*

Alice's Restaurant Property

Proposition

$\forall r, s \geq 0$, $A_{r,s}$ *holds almost surely.*

Proof.

For given r, s and $x_1, \ldots, x_r$, $y_1, \ldots, y_s$, let Noz be the event "there is no z satisfying …". It is easy to see that
$\mathbb{P}\left(\text{Noz}\right) = \left(1 - 2^{-r-s}\right)^{n-r-s}$.

Alice's Restaurant Property

Proposition

$\forall r, s \geq 0$, $A_{r,s}$ holds almost surely.

Proof.

For given r, s and $x_1, \ldots, x_r, y_1, \ldots, y_s$, let Noz be the event "there is no z satisfying …". It is easy to see that $\mathbb{P}(\text{Noz}) = (1 - 2^{-r-s})^{n-r-s}$. The union bound gives the following:

$$\mathbb{P}(\neg A_{r,s}) \leq \binom{n}{r}\binom{n-r}{s}\left(1 - 2^{-r-s}\right)^{n-r-s} \to 0$$

□

Alice's Restaurant Property

Definition

A graph is said to have the Alice's Restaurant Property if it satisfies $A_{r,s}$ for all $r, s \geq 0$.

Alice's Restaurant Property

Definition

A graph is said to have the Alice's Restaurant Property if it satisfies $A_{r,s}$ for all $r, s \geq 0$.

Theorem

There is a unique graph G (up to isomorphism) for which $G \models ARP$.

Alice's Restaurant Property

Proof of existence.

The theory generated by $A_{r,s}$ is partial to the almost sure theory, so it is consistent. Hence, by Gödel's completeness theorem it has a countable or finite model (in our case, countable). □

Alice's Restaurant Property

Proof of existence.

The theory generated by $A_{r,s}$ is partial to the almost sure theory, so it is consistent. Hence, by Gödel's completeness theorem it has a countable or finite model (in our case, countable). □

Proof of uniqueness.

On the board! □

Proof of Fagin's Theorem

Proof.

Consider the theory T generated by $A_{r,s}$ for all $r, s \geq 0$. We have shown that this theory has a unique countable model. Hence, by a previous theorem T is complete. Let B be a first-order sentence. Suppose $T \models B$. By compactness we can derive B from a finite subset of T, say X_i, $i \in [m]$.

Proof of Fagin's Theorem

Proof.

Consider the theory T generated by $A_{r,s}$ for all $r, s \geq 0$. We have shown that this theory has a unique countable model. Hence, by a previous theorem T is complete. Let B be a first-order sentence. Suppose $T \models B$. By compactness we can derive B from a finite subset of T, say X_i, $i \in [m]$. But:

$$\lim_{n\to\infty} \mathbb{P}(\neg B) \leq \lim_{n\to\infty} \sum_{i=1}^{m} \mathbb{P}(\neg X_i) = \sum_{i=1}^{m} \lim_{n\to\infty} \mathbb{P}(\neg X_i) = 0$$

Proof of Fagin's Theorem

Proof.

Consider the theory T generated by $A_{r,s}$ for all $r, s \geq 0$. We have shown that this theory has a unique countable model. Hence, by a previous theorem T is complete. Let B be a first-order sentence. Suppose $T \models B$. By compactness we can derive B from a finite subset of T, say X_i, $i \in [m]$. But:

$$\lim_{n\to\infty} \mathbb{P}(\neg B) \leq \lim_{n\to\infty} \sum_{i=1}^{m} \mathbb{P}(\neg X_i) = \sum_{i=1}^{m} \lim_{n\to\infty} \mathbb{P}(\neg X_i) = 0$$

Otherwise $T \models \neg B$; switching the roles of B and $\neg B$ yields the desired result. □

Rules

Settings

- Two players: Spoiler and Duplicator

Rules

Settings

- Two players: Spoiler and Duplicator
- A known natural number k which states the length of the game in rounds

Rules

Settings

- Two players: Spoiler and Duplicator
- A known natural number k which states the length of the game in rounds
- A board consists of two distinct graphs G_1 and G_2

Rules

Settings

- Two players: Spoiler and Duplicator
- A known natural number k which states the length of the game in rounds
- A board consists of two distinct graphs G_1 and G_2
- We shall call this game $\mathrm{EHR}(G_1, G_2; k)$

Rules

How the i'th round looks like...

- Consists of two moves: Spoiler's move followed by Duplicator's move

Rules

How the i'th round looks like...

- Consists of two moves: Spoiler's move followed by Duplicator's move
- Spoiler selects a vertex in *any of the graphs*, marking it i

Rules

How the i'th round looks like. . .

- Consists of two moves: Spoiler's move followed by Duplicator's move
- Spoiler selects a vertex in *any of the graphs*, marking it i
- Duplicator selects a vertex *in the other graph*, marking it i

Rules

Who wins?

- Let x_i, y_i be the marked vertices of G_1, G_2 respectively, indexed according to their marking order.

Rules

Who wins?

- Let x_i, y_i be the marked vertices of G_1, G_2 respectively, indexed according to their marking order.
- Duplicator wins if for all $i, j \in [k]$, $x_i \sim x_j \iff y_i \sim y_j$.

Rules

Who wins?

- Let x_i, y_i be the marked vertices of G_1, G_2 respectively, indexed according to their marking order.
- Duplicator wins if for all $i, j \in [k]$, $x_i \sim x_j \iff y_i \sim y_j$.
- Otherwise Spoiler wins.

Rules

Who wins?

- Let x_i, y_i be the marked vertices of G_1, G_2 respectively, indexed according to their marking order.
- Duplicator wins if for all $i, j \in [k]$, $x_i \sim x_j \iff y_i \sim y_j$.
- Otherwise Spoiler wins.
- We say that $\mathrm{EHR}(G_1, G_2; k)$ is a win for Duplicator if with a perfect play she wins.

Rules

Who wins?

- Let x_i, y_i be the marked vertices of G_1, G_2 respectively, indexed according to their marking order.
- Duplicator wins if for all $i, j \in [k]$, $x_i \sim x_j \iff y_i \sim y_j$.
- Otherwise Spoiler wins.
- We say that $\mathrm{EHR}(G_1, G_2; k)$ is a win for Duplicator if with a perfect play she wins.

Observation

If G_1, G_2 satisfy Alice's Restaurant Property, Duplicator wins $\mathrm{EHR}(G_1, G_2; k)$ for any k.

Equivalence Classes

Definition

Given G_1, G_2 and a non-negative integer k, we say $(G_1; x_1, \ldots, x_s) \equiv_k (G_2; y_1, \ldots, y_s)$ whenver Duplicator has a winning strategy on the Ehrenfeucht game played on G_1, G_2, assuming the first s moves out of k done, having marked x_i, y_i.

Equivalence Classes

Observation 1

If $s = k$ the game is over, and Duplicator wins exactly if $x_i \sim x_j \iff y_i \sim y_j$.

Equivalence Classes

Observation 1

If $s = k$ the game is over, and Duplicator wins exactly if $x_i \sim x_j \iff y_i \sim y_j$.

Observation 2

If $s = 0$ we obtain our original game. We write: $G_1 \equiv G_2$ if it is a win for Duplicator.

Equivalence Classes

Proposition

For each k, $\equiv_k$ is an equivalence relation.

Equivalence Classes

Proposition

For each k, $\equiv_k$ is an equivalence relation.

Proof of reflexivity.

Indeed, by duplicating Spoiler's moves, Duplicator wins. □

Equivalence Classes

Proposition

For each k, $\equiv_k$ is an equivalence relation.

Proof of reflexivity.

Indeed, by duplicating Spoiler's moves, Duplicator wins. □

Proof of symmetricity.

The order of the graphs plays no role in the game. □

Equivalence Classes

Proof of transitivity.

By reverse induction on s. If $s = k$ from earlier observation we conclude that $x_i \sim x_j \iff y_i \sim y_j \iff z_i \sim z_j$.

Equivalence Classes

Proof of transitivity.

By reverse induction on s. If $s = k$ from earlier observation we conclude that $x_i \sim x_j \iff y_i \sim y_j \iff z_i \sim z_j$. Assume the result for $s + 1$, and consider the game on G_1, G_3 where $(G_1; x_1, \ldots, x_s) \equiv_k (G_2; y_1, \ldots, y_s) \equiv_k (G_3; z_1, \ldots, z_s)$. It is now Spoiler's move, and he marks x_{s+1}. Duplicator has a winning reply in G_2, say y_{s+1}, so $(G_1; x_1, \ldots, x_{s+1}) \equiv_k (G_2; y_1, \ldots, y_{s+1})$. Had Spoiler chosen y_{s+1} in the game G_2, G_3, Duplicator would have had a winning reply in G_3, say z_{s+1}. Hence $(G_2; y_1, \ldots, y_{s+1}) \equiv_k (G_3; z_1, \ldots, z_{s+1})$. Duplicator replies to Spoiler's x_{s+1} by marking z_{s+1}, and wins by induction. □

Combinatorialists like games. Logicians like truth. Fortunately, there is a connection.

-Joel Spencer

The Logic Behind the Game

Theorem

1. *$G_1 \equiv_k G_2$ iff G_1, G_2 agree on all first-order sentences of quantifier depth k.*
2. *For each equivalence class $[G]_{\equiv_k}$ there exists a first-order sentence A of quantifier depth k for which $[G]_{\equiv_k} = \{G' \mid G' \models A\}$.*

The Logic Behind the Game

Theorem (stronger)

For each $k \geq 1$ and $0 \leq s \leq k$

1. $(G_1; x_1, \ldots, x_s) \equiv_k (G_2; y_1, \ldots, y_s)$ *iff G_1, G_2 agree on all first-order predicates of quantifier depth $k - s$ with s free variables, when we assign $x_1, \ldots, x_s$ or $y_1, \ldots, y_s$ to these variables.*
2. *For each equivalence class $[(G; x_1, \ldots, x_s)]_{\equiv_k}$ there exists a first-order predicate A of quantifier depth $k - s$ with s free variables, for which* $[(G; x_1, \ldots, x_s)]_{\equiv_k} = \{(G'; y_1, \ldots, y_s) \mid G' \models A(y_1, \ldots, y_s)\}$.

The Logic Behind the Game

Proof of the case $s = k$.

We note that $(G_1; x_1, \ldots, x_k) \equiv_k (G_2; y_1, \ldots, y_k)$ iff the induced subgraphs of G_1, G_2 on their designated vertices are the same. Any predicate of quantifier depth $k - s = 0$ is a boolean combination of $x_i \sim x_j$ and $x_i = x_j$, hence the equivalence implies agreement with regard to such a predicate, while inequivalence implies disagreement with regard to one such predicate.

The Logic Behind the Game

Proof of the case $s = k$.

We note that $(G_1; x_1, \ldots, x_k) \equiv_k (G_2; y_1, \ldots, y_k)$ iff the induced subgraphs of G_1, G_2 on their designated vertices are the same. Any predicate of quantifier depth $k - s = 0$ is a boolean combination of $x_i \sim x_j$ and $x_i = x_j$, hence the equivalence implies agreement with regard to such a predicate, while inequivalence implies disagreement with regard to one such predicate.
The predicate A that lists the adjacencies and nonadjacencies amongst the x_i's will be the one to define $[(G_1; x_1, \ldots, x_k)]_{\equiv_k}$.

The Logic Behind the Game

Proof of the case $s < k$, assuming correctness for $s + 1$

From induction, each β of the form $[(G'; y_1, \dots, y_s, y_{s+1})]_{\equiv_k}$ is defined by a predicate A_β of quantifier depth $k - s - 1$, having $s + 1$ free variables. Let $\alpha = [(G; x_1, \dots, x_s)]_{\equiv_k}$ and let $\overline{\alpha}$ be the representative $(G; x_1, \dots, x_s)$. Define $\varphi(\beta) = \exists x A_\beta (x_1, \dots, x_s, x)$. Define also $\text{Yes}[\overline{\alpha}] = \{\beta \mid \overline{\alpha} \models \varphi(\beta)\}$ and $\text{No}[\overline{\alpha}] = \{\beta \mid \overline{\alpha} \models \neg\varphi(\beta)\}$. We will later show that these sets do not depend on the representative $\overline{\alpha}$, hence we can mark them $\text{Yes}[\alpha]$ and $\text{No}[\alpha]$. We define $A_\alpha = \bigwedge_{\beta \in \text{Yes}[\alpha]} \varphi(\beta) \wedge \bigwedge_{\beta \in \text{No}[\alpha]} \neg\varphi(\beta)$.

The Logic Behind the Game

Proof (cont.) — why A_α works?

First we note that A_α is of quantifier depth $k - s$ and with s free variables, as wanted. Clearly, $\alpha \models A_\alpha$. Suppose $\overline{\gamma} \models A_\alpha$. The set of equivalence classes generated by $\overline{\gamma}$ with an additional designated x is exactly Yes $[\alpha]$, hence $\overline{\gamma} \in \alpha$.

The Logic Behind the Game

Proof (cont.) — why the representative does not matter?

Suppose $\alpha_1, \alpha_2 \in \alpha$, two representatives. Assume $\beta \in \text{Yes}\,[\alpha_1]$. Hence $\alpha_1 \models \exists x A_\beta\,(x_1, \ldots, x_s, x)$. We want to show that $\alpha_2 \models \exists x A_\beta\,(y_1, \ldots, y_s, x)$. Indeed, $(G_1; x_1, \ldots, x_s) \equiv_k (G_2; y_1, \ldots, y_s)$, hence $(G_1; x_1, \ldots, x_s, z)$ models A_β form some z. Let z' be the winning reply of Duplicator to z on the $\text{EHR}(G_1, G_2; k)$ game. Hence $(G_1; x_1, \ldots, x_s, z) \equiv_k (G_2; y_1, \ldots, y_s, z')$, hence by induction $(G_2; y_1, \ldots, y_s, z')$ models A_β, hence $\alpha_2 \models \exists x A_\beta\,(y_1, \ldots, y_s, x)$, as wanted.

The Logic Behind the Game

Proof (cont.) — proving the first part of the theorem.

Suppose G_1, G_2 (with designated vertices) agree on first-order prediacates of quantifier depth $k - s$ with s free variables. Hence, they agree on the same predicate that defines the equivalence class of G_1, hence they are equivalent.

The Logic Behind the Game

Proof (cont.) — proving the first part of the theorem.

Suppose G_1, G_2 (with designated vertices) agree on first-order prediacates of quantifier depth $k-s$ with s free variables. Hence, they agree on the same predicate that defines the equivalence class of G_1, hence they are equivalent.

Conversely, let G_1, G_2 (with designated vertices) be k-equivalent, and let P be some predicate of quantifier depth $k-s$ and s free variables. We can express P is a boolean combination of phrases of the form $\exists x Q$ where Q is of quantifier depth $k-s-1$ and $s+1$ free variables. By induction, the value of Q is determined by the equivalence class of $(G; x_1, \ldots, x_s, x)$ for every x, hence the value of P is determined by the equivalence class of $(G; x_1, \ldots, x_s)$, hence G_1, G_2 agree on P. □

Examples

Theorem

Connectivity is not first-order expressible.

Examples

Theorem

Connectivity is not first-order expressible.

Proof sketch.

We let G_1 be a cycle of length n and G_2 be two such cycles, with n at least 2^k. With s moves remaining in the game, Duplicator calls any two vertices of distance at most 2^s "close enough", and do her best to reply in a way that the corresponding points will be of the same distance apart and the same orientation on the other graph.

Examples

Theorem

Connectivity is not first-order expressible.

Proof sketch.

We let G_1 be a cycle of length n and G_2 be two such cycles, with n at least 2^k. With s moves remaining in the game, Duplicator calls any two vertices of distance at most 2^s "close enough", and do her best to reply in a way that the corresponding points will be of the same distance apart and the same orientation on the other graph. When Spoiler tries to take advantage of G_2's nonconnectivity, Duplicator replies with marking vertices so far apart, so that Spoiler will not have time to spoil. □

Examples

Theorem

2-colourability is not first-order expressible.

Examples

Theorem

2-colourability is not first-order expressible.

Proof sketch.

We take G_1 to be a cycle of length $2n$ and G_2 to be a cycle of length $2n+1$, for large enough n, and use a similar argument. □

Thank You!

☺