Mathematical Preliminaries

Mathematical Preliminaries

- Sets
- Functions
- Relations
- Graphs
- Proof Techniques

SETS

A set is a collection of elements

$$A = \{1, 2, 3\}$$

$$B = \{train, bus, bicycle, airplane\}$$

We write

$$1 \in A$$

$$ship \notin B$$

Set Representations

$C = \{ a, b, c, d, e, f, g, h, i, j, k \}$

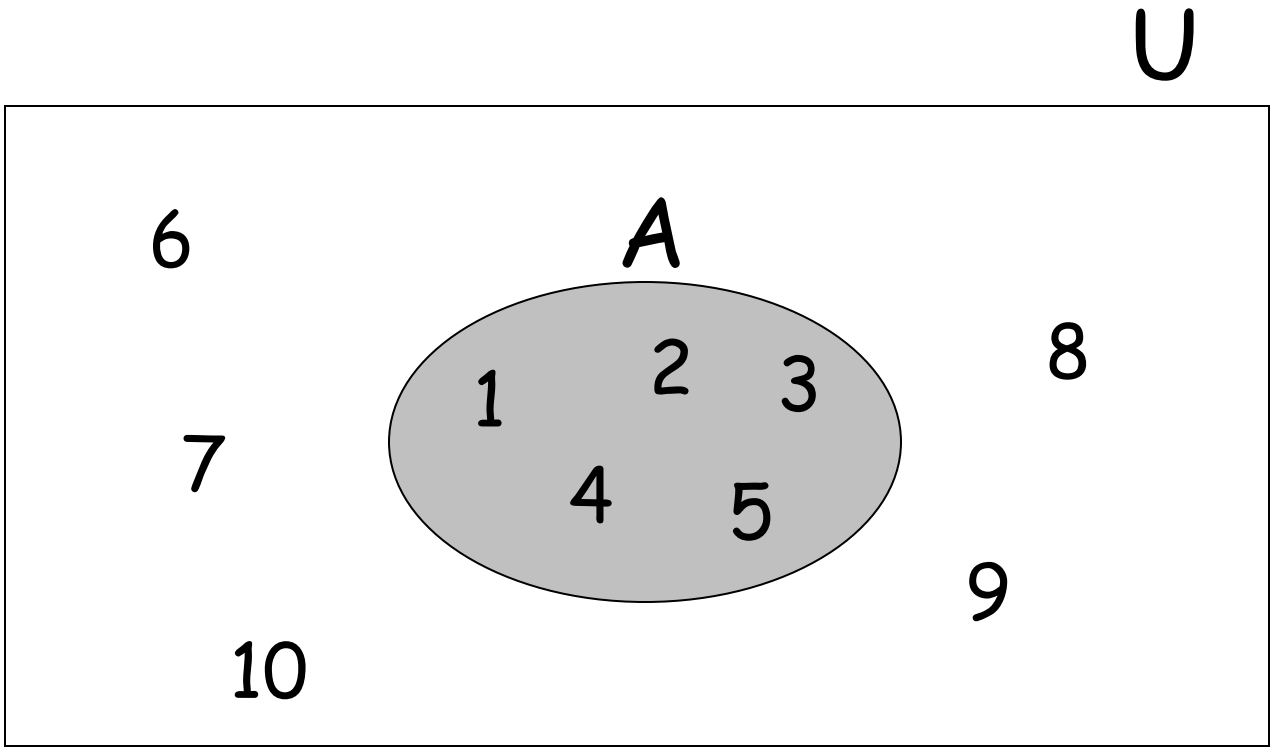
$C = \{ a, b, \ldots, k \}$ ⟶ *finite set*

$S = \{ 2, 4, 6, \ldots \}$ ⟶ *infinite set*

$S = \{ j : j > 0, \text{ and } j = 2k \text{ for some } k>0 \}$

$S = \{ j : j \text{ is nonnegative and even} \}$

$A = \{ 1, 2, 3, 4, 5 \}$



Universal Set: all possible elements

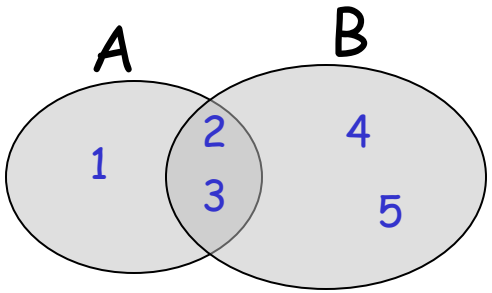
$U = \{ 1 , \dots , 10 \}$

Set Operations

$A = \{ 1, 2, 3 \}$ $B = \{ 2, 3, 4, 5\}$

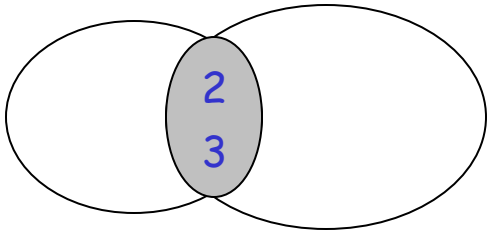
- Union

$A \cup B = \{ 1, 2, 3, 4, 5 \}$



- Intersection

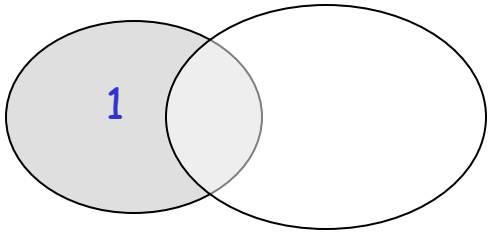
$A \cap B = \{ 2, 3 \}$



- Difference

$A - B = \{ 1 \}$

$B - A = \{ 4, 5 \}$

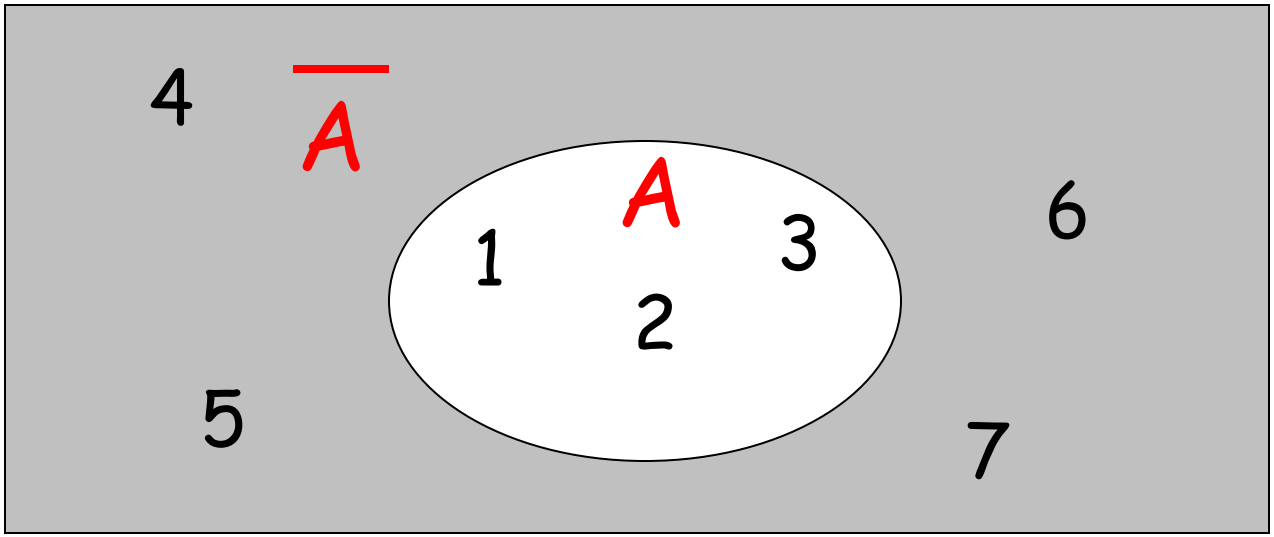


Venn diagrams

- Complement

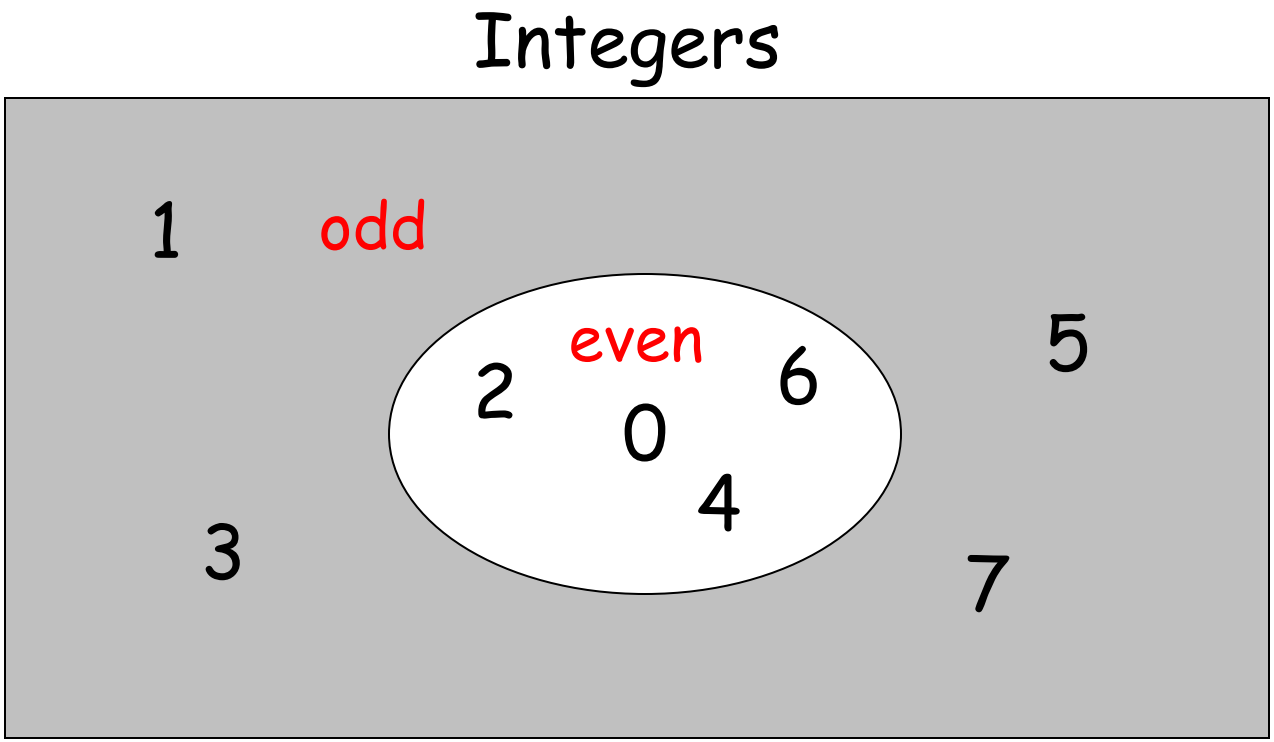
Universal set = {1, …, 7}

$A = \{ 1, 2, 3 \} \Rightarrow \overline{A} = \{ 4, 5, 6, 7 \}$



$$\overline{\overline{A}} = A$$

$$\overline{\{\text{ even integers }\}} = \{\text{ odd integers }\}$$



DeMorgan's Laws

$$\overline{A \cup B} = \overline{A} \cap \overline{B}$$

$$\overline{A \cap B} = \overline{A} \cup \overline{B}$$

Empty, Null Set: $\emptyset$

$\emptyset = \{\}$

$S \cup \emptyset = S$

$S \cap \emptyset = \emptyset$

$S - \emptyset = S$

$\emptyset - S = \emptyset$

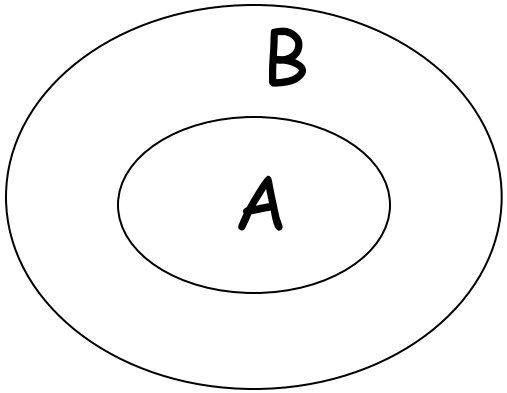
$\overline{\emptyset}$ = Universal Set

Subset

$A = \{ 1, 2, 3\}$ $B = \{ 1, 2, 3, 4, 5 \}$

$A \subseteq B$

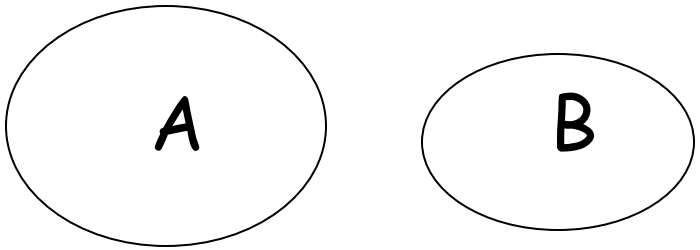
Proper Subset: $A \subset B$



Disjoint Sets

$A = \{ 1, 2, 3 \}$ $B = \{ 5, 6\}$

$A \cap B = \emptyset$



Set Cardinality

- For finite sets

$A = \{ 2, 5, 7 \}$

$|A| = 3$

(set size)

Powersets

A powerset is a set of sets

$S = \{ a, b, c \}$

Powerset of S = the set of all the subsets of S

$2^S = \{ \emptyset, \{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\} \}$

Observation: $| 2^S | = 2^{|S|}$ $\quad (8 = 2^3)$

Cartesian Product

$A = \{ 2, 4 \}$ $B = \{ 2, 3, 5 \}$

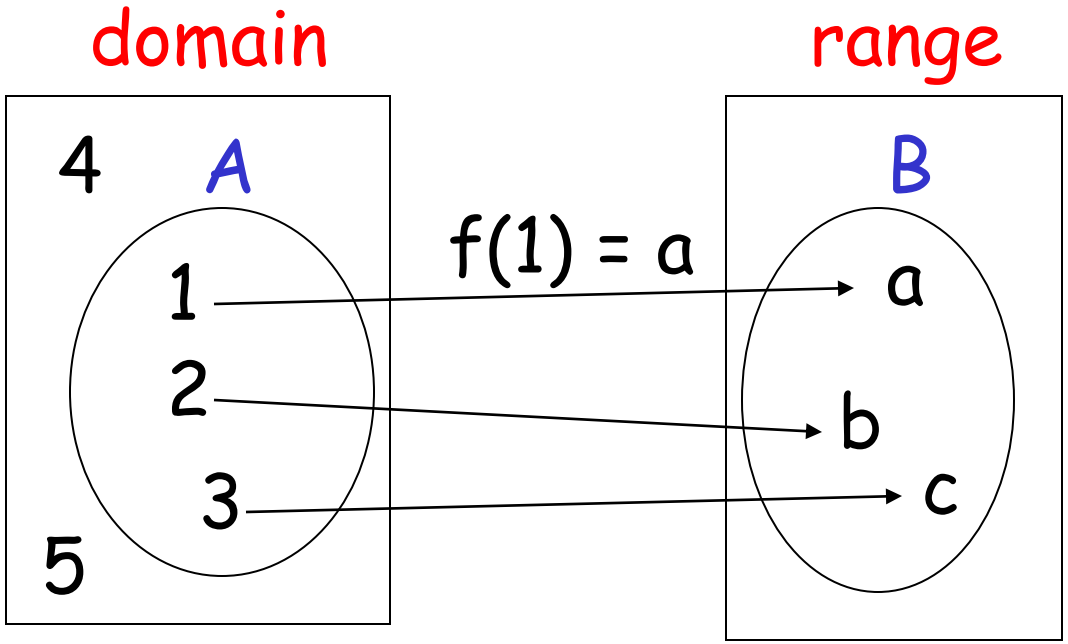
$A \times B = \{ (2, 2), (2, 3), (2, 5), (4, 2), (4, 3), (4, 5) \}$

$|A \times B| = |A| \ |B|$

Generalizes to more than two sets

$A \times B \times \ldots \times Z$

Functions



f : A -> B

If A = domain

then f is a total function

otherwise f is a partial function

Relations

$R = \{(x_1, y_1), (x_2, y_2), (x_3, y_3), \ldots\}$

$x_i \; R \; y_i$

e. g. if R = '>': $2 > 1$, $3 > 2$, $3 > 1$

Equivalence Relations

- Reflexive: $x \; R \; x$
- Symmetric: $x \; R \; y \Rightarrow y \; R \; x$
- Transitive: $x \; R \; y$ and $y \; R \; z \Rightarrow x \; R \; z$

Example: R = '='

- $x = x$
- $x = y \Rightarrow y = x$
- $x = y$ and $y = z \Rightarrow x = z$

Equivalence Classes

For equivalence relation R

equivalence class of $x = \{y : x \, R \, y\}$

Example:

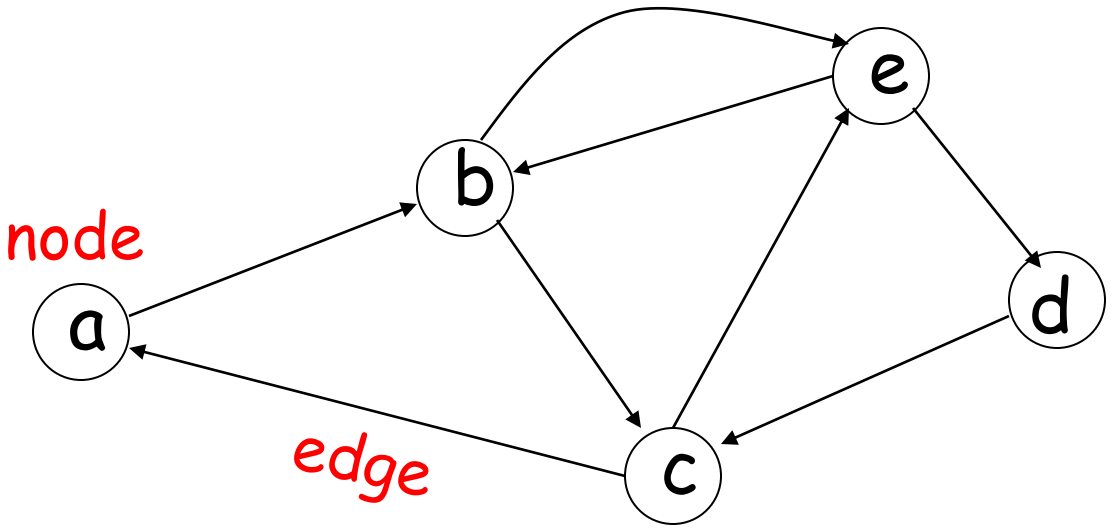
$R = \{ (1, 1), (2, 2), (1, 2), (2, 1),$
$(3, 3), (4, 4), (3, 4), (4, 3) \}$

Equivalence class of $1 = \{1, 2\}$

Equivalence class of $3 = \{3, 4\}$

Graphs

A directed graph



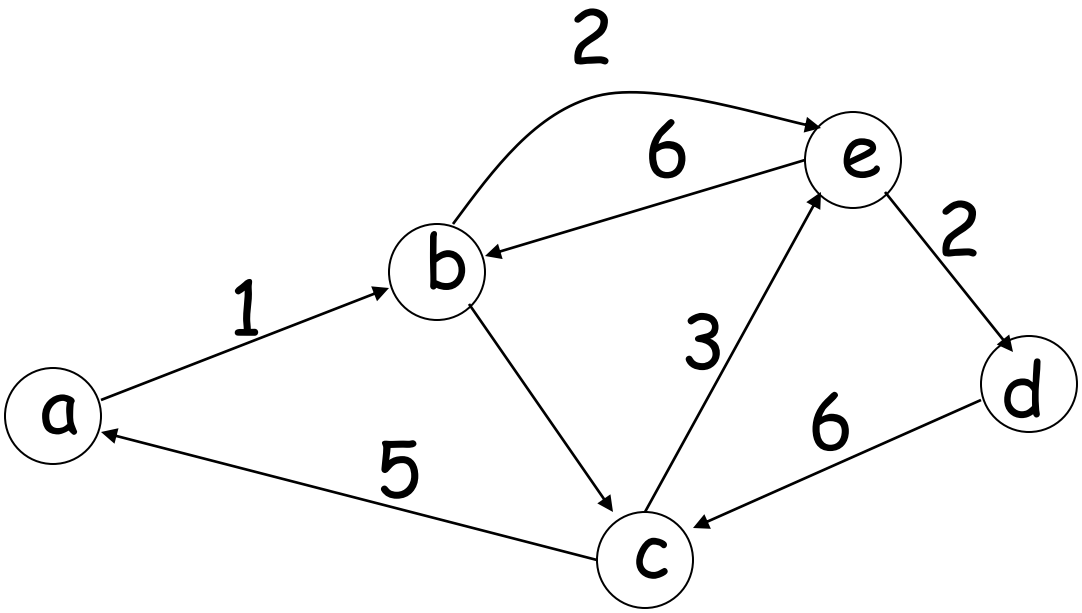
- Nodes (Vertices)

$$V = \{ a, b, c, d, e \}$$

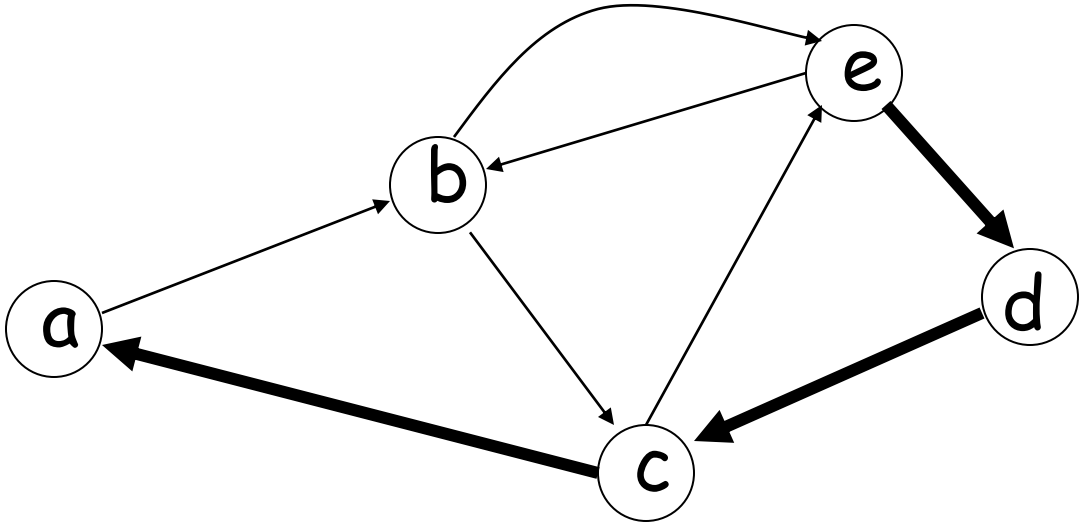
- Edges

$$E = \{ (a,b), (b,c), (b,e), (c,a), (c,e), (d,c), (e,b), (e,d) \}$$

Labeled Graph



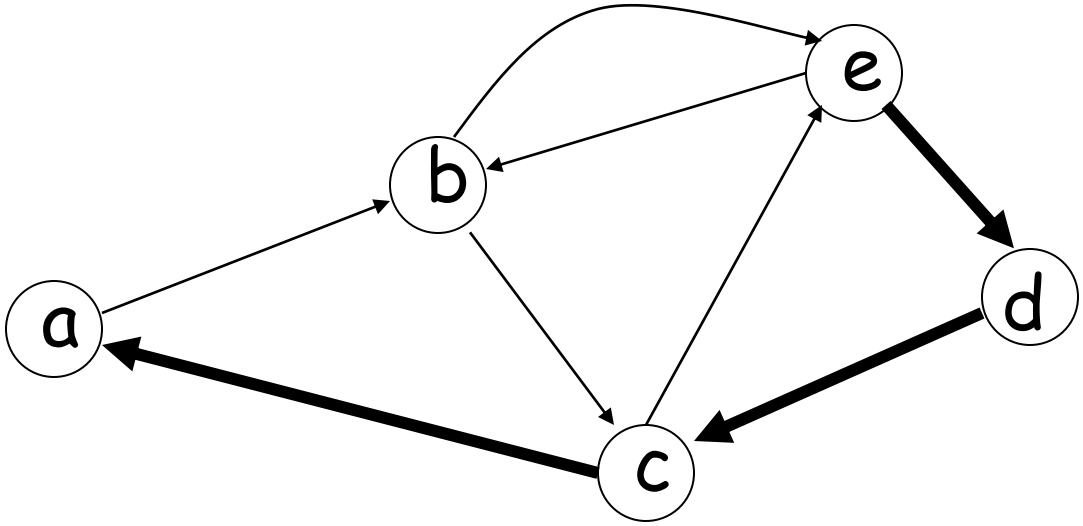
Walk



Walk is a sequence of adjacent edges

(e, d), (d, c), (c, a)

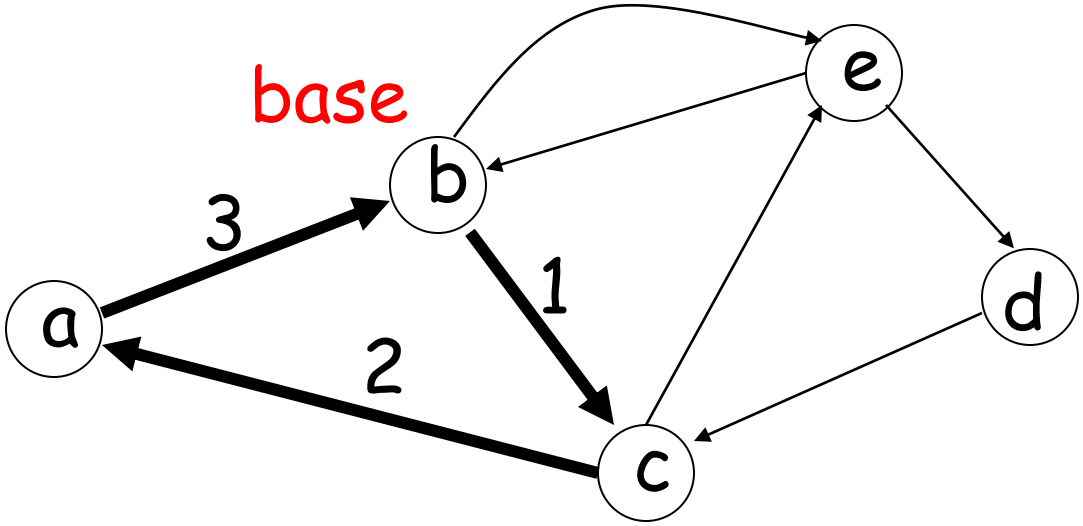
Path



Path is a walk where no edge is repeated

Simple path: no node is repeated

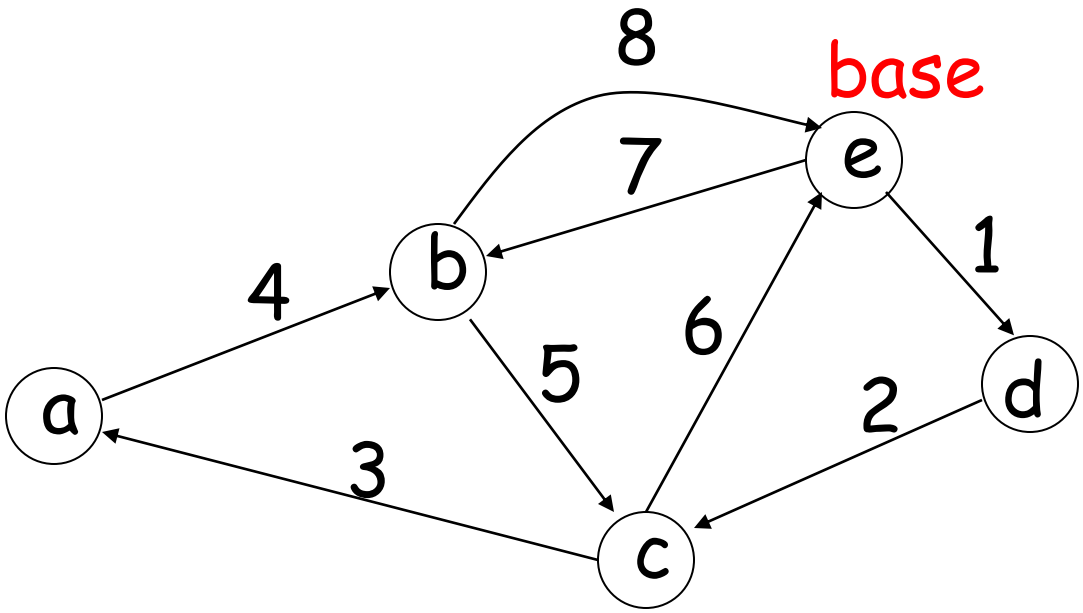
Cycle



Cycle: a walk from a node (base) to itself

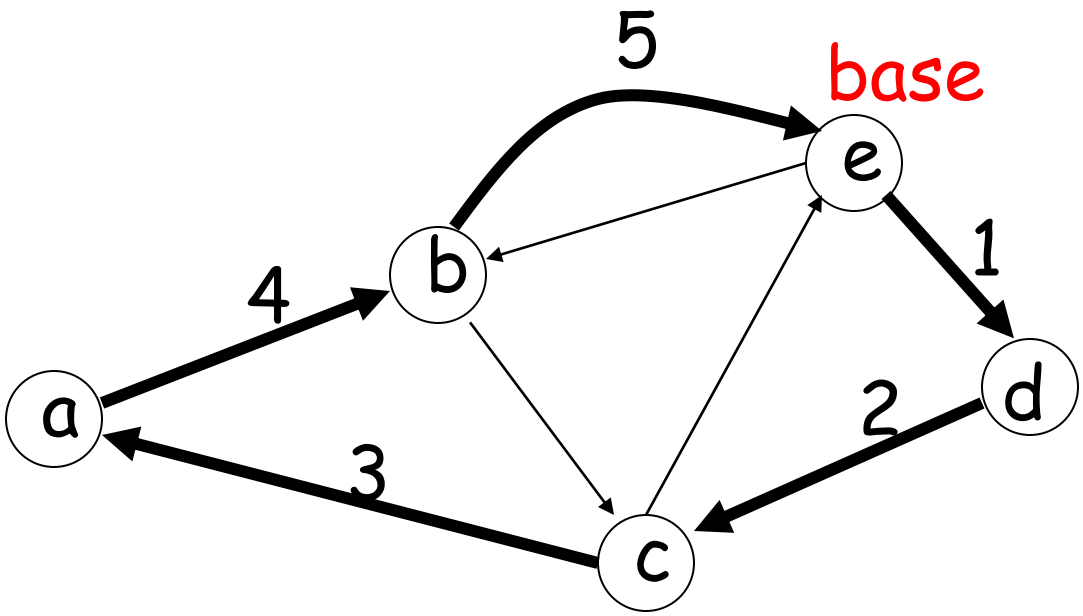
Simple cycle: only the base node is repeated

Euler Tour



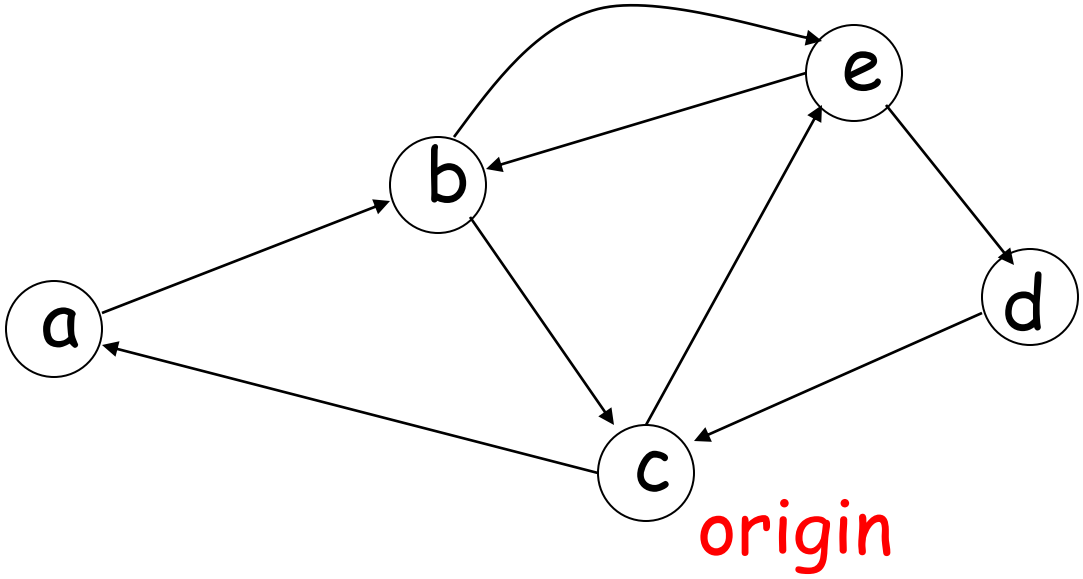
A cycle that contains each edge once

Hamiltonian Cycle

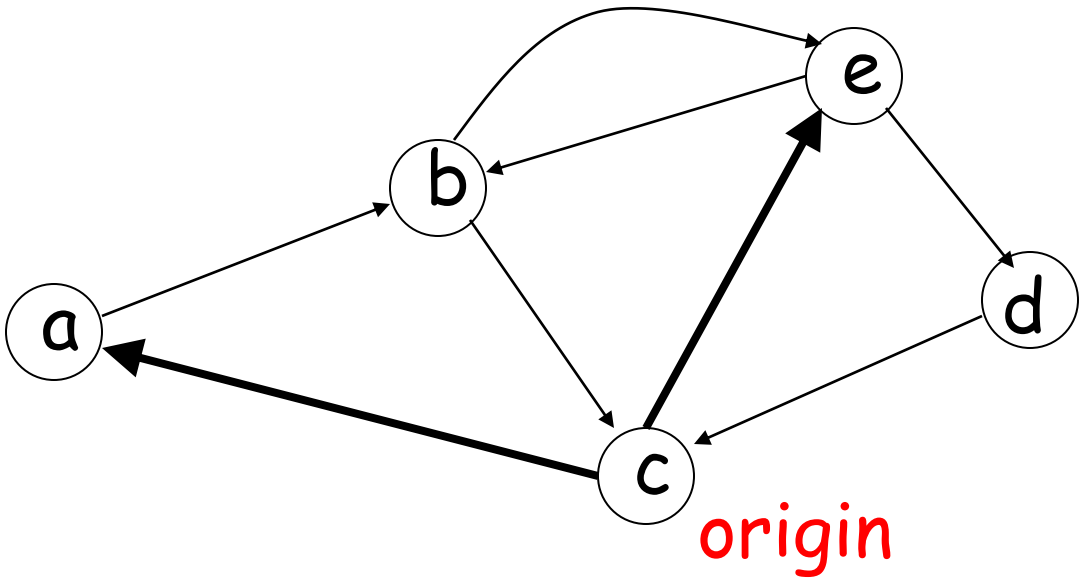


A simple cycle that contains all nodes

Finding All Simple Paths



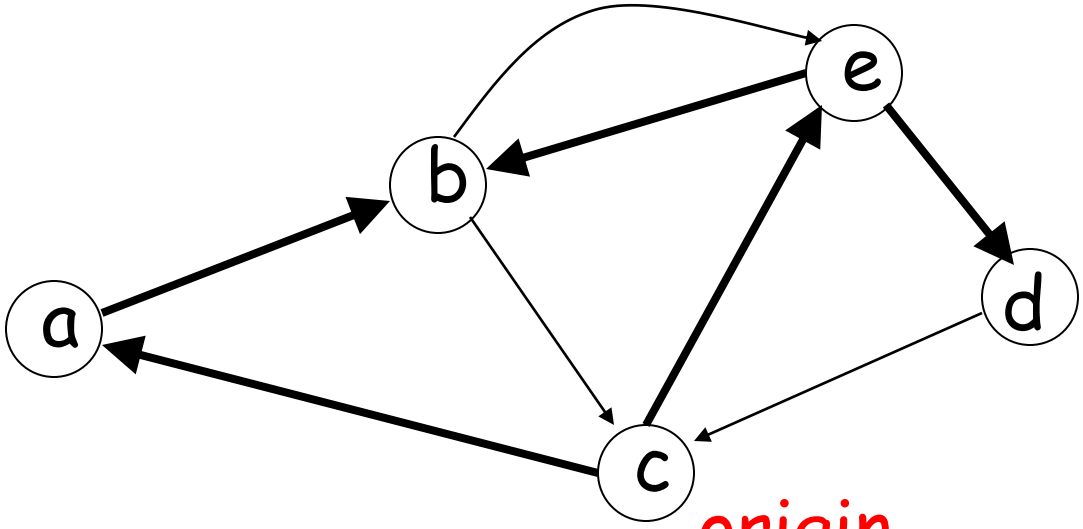
Step 1



(c, a)

(c, e)

Step 2



(c, a)

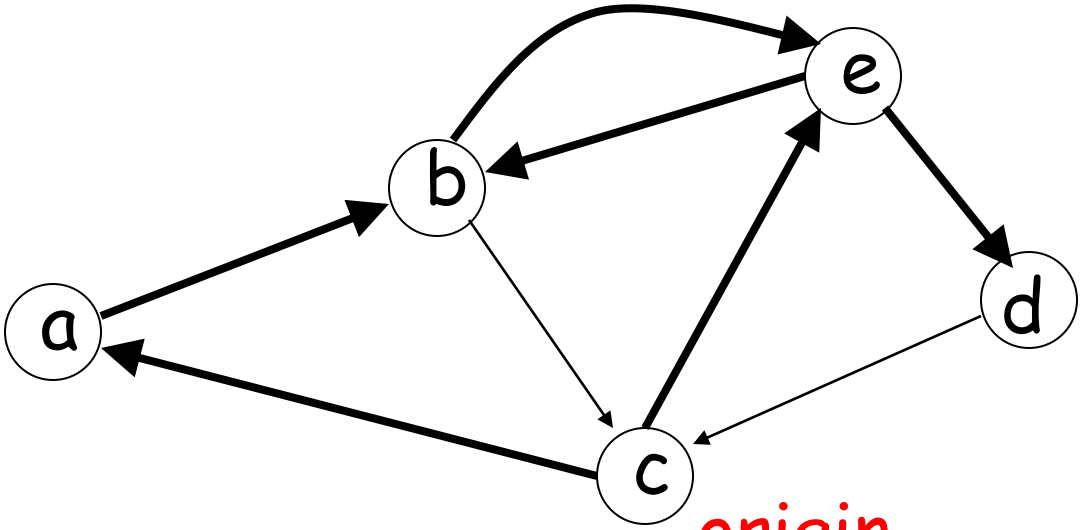
(c, a), (a, b)

(c, e)

(c, e), (e, b)

(c, e), (e, d)

Step 3



(c, a)

(c, a), (a, b)

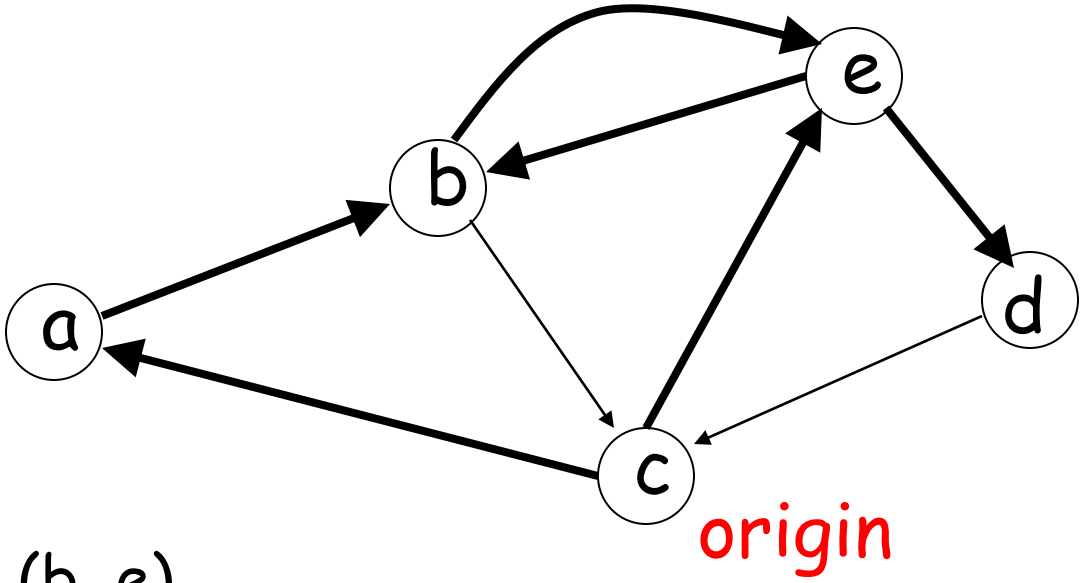
(c, a), (a, b), (b, e)

(c, e)

(c, e), (e, b)

(c, e), (e, d)

Step 4



(c, a)

(c, a), (a, b)

(c, a), (a, b), (b, e)

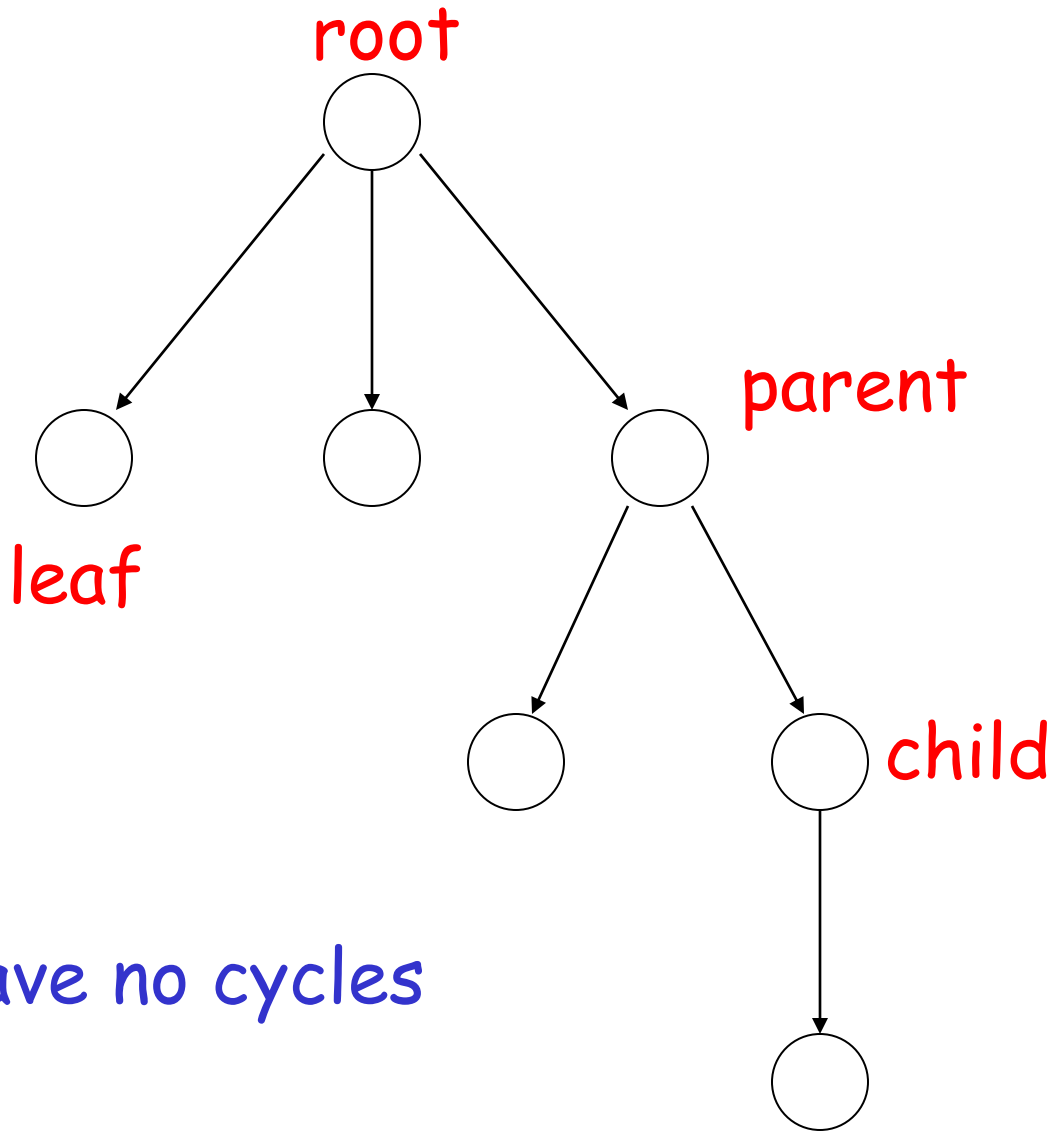
(c, a), (a, b), (b, e), (e,d)

(c, e)

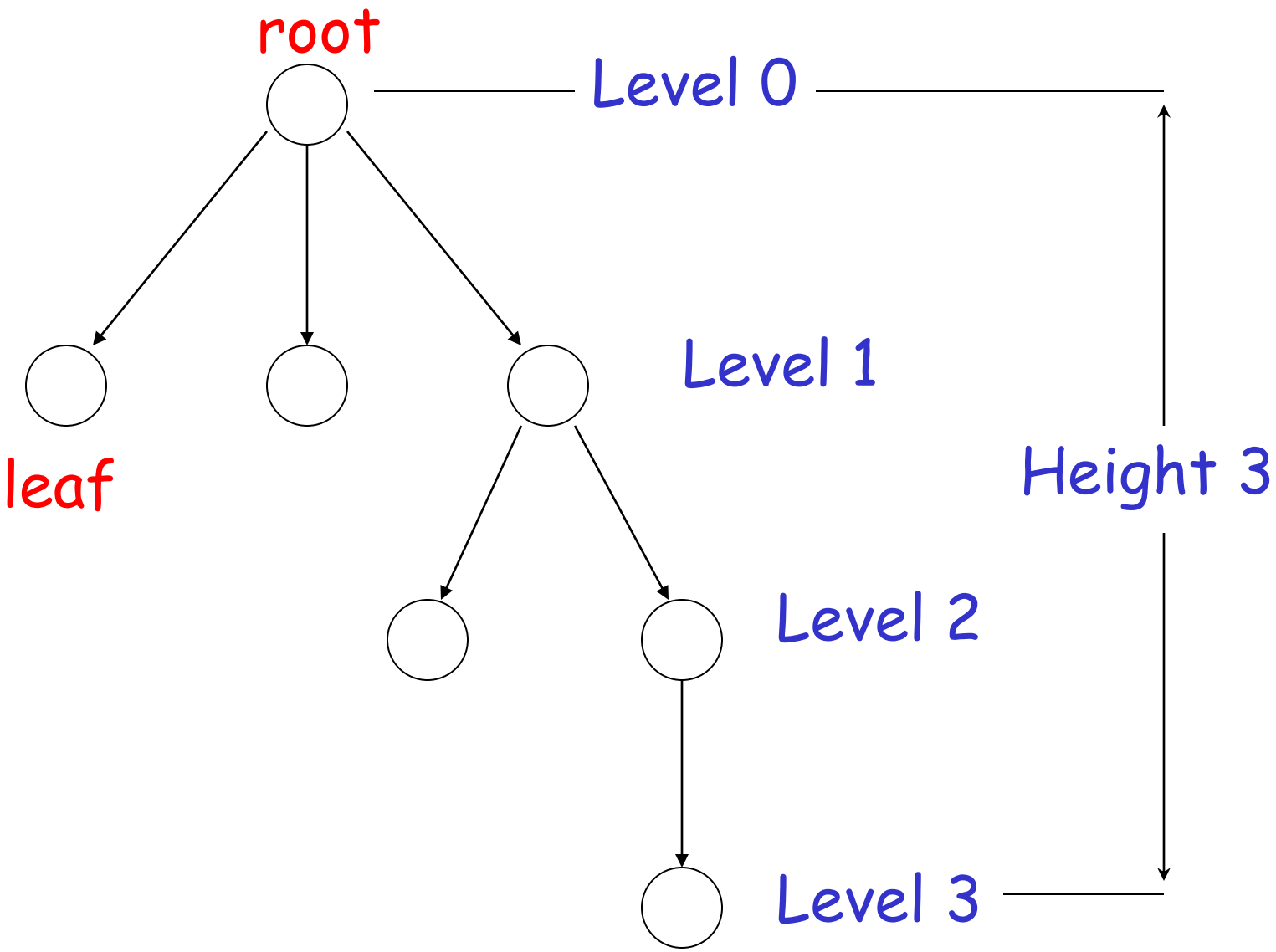
(c, e), (e, b)

(c, e), (e, d)

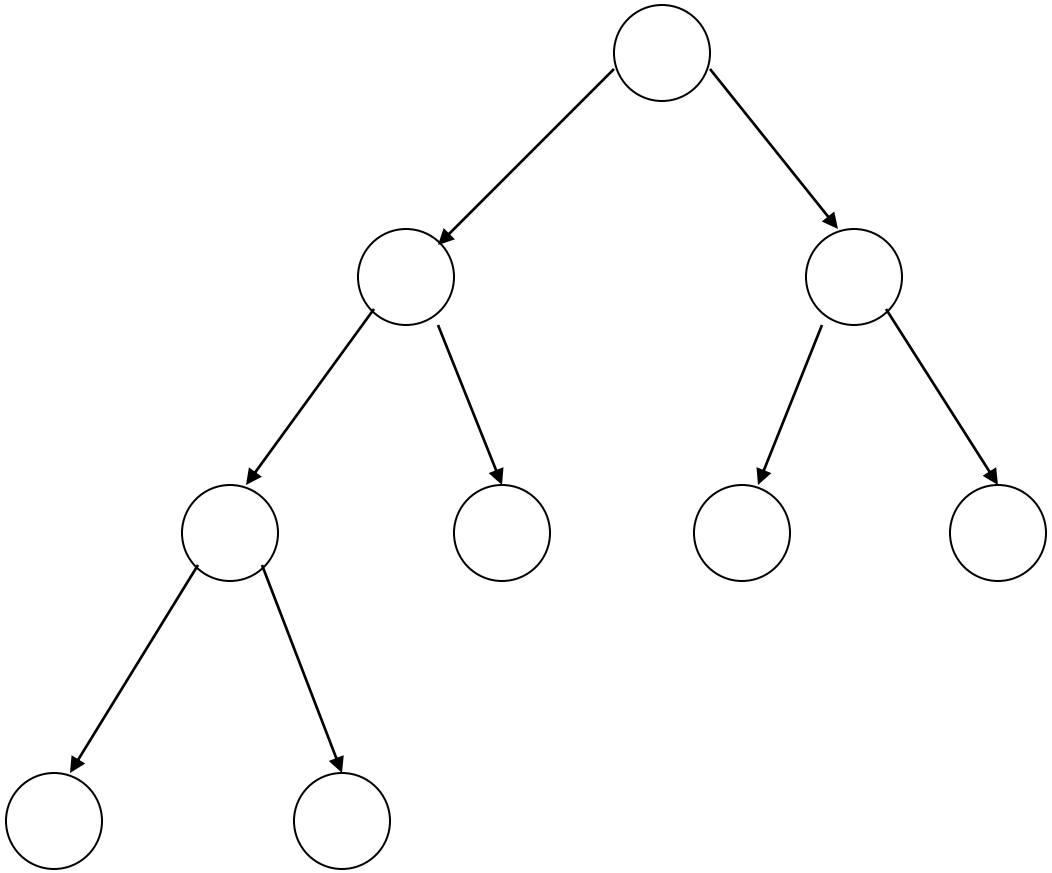
Trees



Trees have no cycles


root
Level 0
Level 1
leaf
Height 3
Level 2
Level 3

Binary Trees



Proof Techniques

- Proof by induction
- Proof by contradiction
- Proof by construction

Induction

We have statements $P_1, P_2, P_3, \ldots$

If we know

- for some b that $P_1, P_2, \ldots, P_b$ are true
- for any k >= b that

$P_1, P_2, \ldots, P_k$ imply P_{k+1}

Then

Every P_i is true

Proof by Induction

- Inductive basis

  Find $P_1, P_2, \ldots, P_b$ which are true

- Inductive hypothesis

  Let's assume $P_1, P_2, \ldots, P_k$ are true,

  for any k >= b

- Inductive step

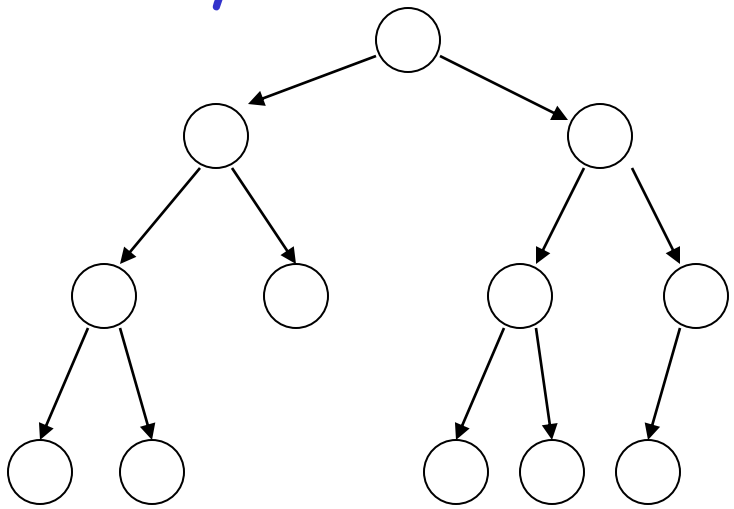
  Show that P_{k+1} is true

Example

Theorem: A binary tree of height n has at most 2^n leaves.

Proof by induction:

let L(i) be the maximum number of leaves of any subtree at height i



We want to show: $L(i) <= 2^i$

- Inductive basis

  $L(0) = 1$ (the root node) ○

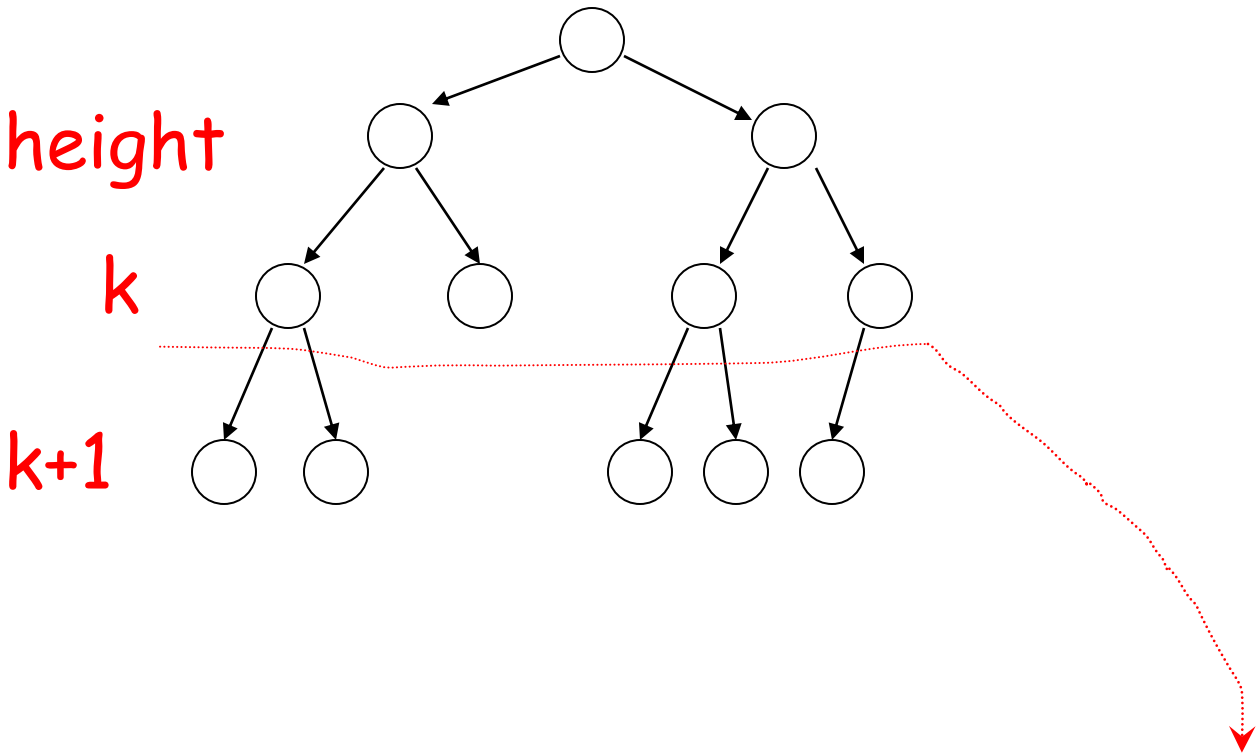
- Inductive hypothesis

  Let's assume $L(i) <= 2^i$ for all $i = 0, 1, \ldots, k$

- Induction step

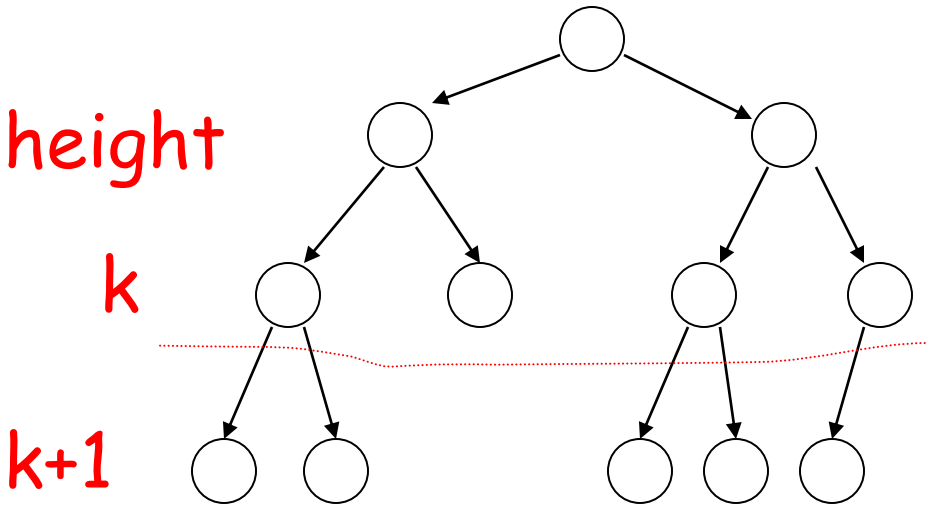
  we need to show that $L(k + 1) <= 2^{k+1}$

Induction Step



From Inductive hypothesis: $L(k) <= 2^k$

Induction Step



$L(k) <= 2^k$

$L(k+1) <= 2 * L(k) <= 2 * 2^k = 2^{k+1}$

(we add at most two nodes for every leaf of level k)

Remark

Recursion is another thing

Example of recursive function:

$$f(n) = f(n-1) + f(n-2)$$

$$f(0) = 1, \quad f(1) = 1$$

Proof by Contradiction

We want to prove that a statement P is true

- we assume that P is false
- then we arrive at an incorrect conclusion
- therefore, statement P must be true

Example

Theorem: $\sqrt{2}$ is not rational

Proof:

Assume by contradiction that it is rational

$\sqrt{2} = n/m$

n and m have no common factors

We will show that this is impossible

$\sqrt{2} = n/m$ ⇒ $2\,m^2 = n^2$

Therefore, n^2 is even ⇒ n is even

$n = 2\,k$

$2\,m^2 = 4k^2$ ⇒ $m^2 = 2k^2$ ⇒ m is even

$m = 2\,p$

Thus, m and n have common factor 2

Contradiction!

Proof by Construction

We want to prove that a statement about something with a property is true

- constructing a concrete example with a property to show that something having that property exists.
- constructive proof is in contrast to a non-constructive proof which does not provide a means of constructing an example.

Example 1

16 can be exactly divided.

Proof

- A concrete example is 16/2. Therefore, the statement is true.

End

Example 2

There exist two irrational numbers which make a^b rational.

Proof

Let a=b= $\sqrt{2}$

case 1: $\sqrt{2}^{\sqrt{2}}$ is rational. done, otherwise

case 2: let a= $\sqrt{2}^{\sqrt{2}}$, then $(\sqrt{2}^{\sqrt{2}})^{\sqrt{2}} = \sqrt{2}^{2} = 2$, done.

End

Question

Is example 2 the constructive proof?

Why if yes? Why if no?

Example 3

Show that there is no "largest integer".

Proof

Let n be any integer.

Let $m = n+1$

m is an integer

$m > n$

Therefore m is an integer that is larger than n

Therefore, for any integer there exists an integer $m = n + 1$ that is larger than it.

End

Question

Is example 3 the constructive proof?

Why if yes? Why if no?

Example 4

Show that there is no "largest" prime number.

Proof

Let n be any prime number

Let $m = n! + 1$, then $m > n$

Case 1:

$m = n! + 1$ is a prime number, then we have constructed a prime number that is larger than the previous prime number.

Case 2:

$m = n! + 1$ is not a prime number, then it has at least one prime factor

Example 4 (Cont.)

Explanation:

If you divide m by any of the prime numbers that are smaller than or equal to n, you will always get a remainder of 1,

because each prime number less than or equal to n divides evenly into n!.

Therefore any prime factors of m must be greater than n.

End

Question

Is example 4 the constructive proof?

Why if yes? Why if no?

Languages

A language is a set of strings

String: A sequence of letters

Examples: "cat", "dog", "house", …

Defined over an alphabet:

$$\Sigma = \{a, b, c, \ldots, z\}$$

Alphabets and Strings

We will use small alphabets: $\Sigma = \{a, b\}$

Strings

a

ab $\quad u = ab$

$abba$ $\quad v = bbbaaa$

$baba$ $\quad w = abba$

$aaabbbaabab$

String Operations

$w = a_1 a_2 \cdots a_n$

$v = b_1 b_2 \cdots b_m$

$abba$

$bbbaaa$

Concatenation

$wv = a_1 a_2 \cdots a_n b_1 b_2 \cdots b_m$ $abbabbbaaa$

$w = a_1 a_2 \cdots a_n$ $ababaaabbb$

Reverse

$w^R = a_n \cdots a_2 a_1$ $bbbaaababa$

String Length

$$w = a_1 a_2 \cdots a_n$$

Length: $|w| = n$

Examples:

$$|abba| = 4$$

$$|aa| = 2$$

$$|a| = 1$$

Length of Concatenation

$$|uv| = |u| + |v|$$

Example: $u = aab, \quad |u| = 3$

$v = abaab, \quad |v| = 5$

$$|uv| = |aababaaab| = 8$$

$$|uv| = |u| + |v| = 3 + 5 = 8$$

Empty String

A string with no letters: λ

Observations: $|\lambda| = 0$

$$\lambda w = w\lambda = w$$

$$\lambda abba = abba\lambda = abba$$

Substring

Substring of string:

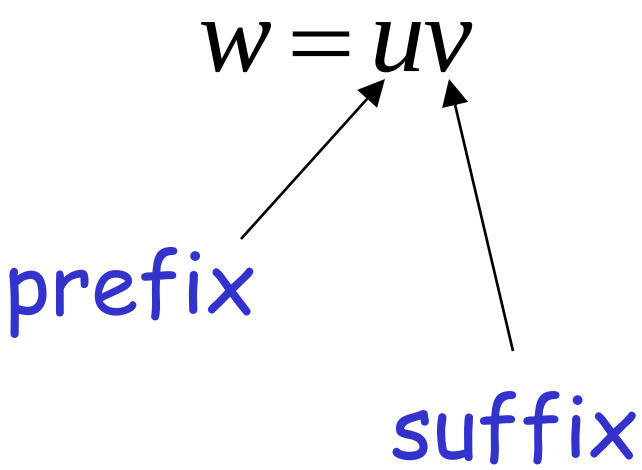
a subsequence of consecutive characters

String	Substring
$ab\underline{bab}$	ab
$ab\underline{bab}$	$abba$
$ab\underline{b}ab$	b
$a\underline{bbab}$	$bbab$

Prefix and Suffix

$abbab$

Prefixes	Suffixes
λ	$abbab$
a	$bbab$
ab	bab
abb	ab
$abba$	b
$abbab$	λ



Another Operation

$$w^n = \underbrace{ww\cdots w}_{n}$$

Example: $(abba)^2 = abbaabba$

Definition: $w^0 = \lambda$

$$(abba)^0 = \lambda$$

The * Operation

Σ^*: the set of all possible strings from alphabet Σ

$$\Sigma = \{a, b\}$$

$$\Sigma^* = \{\lambda, a, b, aa, ab, ba, bb, aaa, aab, \ldots\}$$

The + Operation

Σ^+ : the set of all possible strings from alphabet Σ except λ

$$\Sigma = \{a, b\}$$

$$\Sigma^* = \{\lambda, a, b, aa, ab, ba, bb, aaa, aab, \ldots\}$$

$$\Sigma^+ = \Sigma^* - \lambda$$

$$\Sigma^+ = \{a, b, aa, ab, ba, bb, aaa, aab, \ldots\}$$

Languages

A language is any subset of Σ^*

Example: $\Sigma = \{a, b\}$

$$\Sigma^* = \{\lambda, a, b, aa, ab, ba, bb, aaa, \ldots\}$$

Languages: $\{\lambda\}$

$\{a, aa, aab\}$

$\{\lambda, abba, baba, aa, ab, aaaaaa\}$

Note that:

Sets $\emptyset = \{\} \neq \{\lambda\}$

Set size $|\{\}| = |\emptyset| = 0$

Set size $|\{\lambda\}| = 1$

String length $|\lambda| = 0$

Another Example

An infinite language $L = \{a^n b^n : n \geq 0\}$

λ

ab

$aabb$

$aaaaabbbbb$

$\in L$

$abb \notin L$

Operations on Languages

The usual set operations

$$\{a, ab, aaaa\} \cup \{bb, ab\} = \{a, ab, bb, aaaa\}$$

$$\{a, ab, aaaa\} \cap \{bb, ab\} = \{ab\}$$

$$\{a, ab, aaaa\} - \{bb, ab\} = \{a, aaaa\}$$

Complement: $\overline{L} = \Sigma^* - L$

$$\overline{\{a, ba\}} = \{\lambda, b, aa, ab, bb, aaa, \ldots\}$$

Reverse

Definition: $L^R = \{w^R : w \in L\}$

Examples: $\{ab, aab, baba\}^R = \{ba, baa, abab\}$

$$L = \{a^n b^n : n \geq 0\}$$

$$L^R = \{b^n a^n : n \geq 0\}$$

Concatenation

Definition: $L_1 L_2 = \{xy : x \in L_1, y \in L_2\}$

Example: $\{a, ab, ba\}\{b, aa\}$

$= \{ab, aaa, abb, abaa, bab, baaa\}$

Another Operation

Definition: $L^n = \underbrace{LL\cdots L}_{n}$

$$\{a,b\}^3 = \{a,b\}\{a,b\}\{a,b\} =$$

$$\{aaa, aab, aba, abb, baa, bab, bba, bbb\}$$

Special case: $L^0 = \{\lambda\}$

$$\{a, bba, aaa\}^0 = \{\lambda\}$$

More Examples

$$L = \{a^n b^n : n \geq 0\}$$

$$L^2 = \{a^n b^n a^m b^m : n, m \geq 0\}$$

$$aabbaaabbb \in L^2$$

Star-Closure (Kleene *)

Definition: $L^* = L^0 \bigcup L^1 \bigcup L^2 \cdots$

Example:

$$\{a, bb\}^* = \left\{ \begin{array}{l} \lambda, \\ a, bb, \\ aa, abb, bba, bbbb, \\ aaa, aabb, abba, abbbb, \ldots \end{array} \right\}$$

Positive Closure

Definition: $L^+ = L^1 \bigcup L^2 \bigcup \cdots$

$$= L* - \{\lambda\}$$

$$\{a, bb\}^+ = \left\{ \begin{array}{l} a, bb, \\ aa, abb, bba, bbbb, \\ aaa, aabb, abba, abbbb, \ldots \end{array} \right\}$$

The End