Kræsjkurs MNF130

Logikk - Definisjoner

Hva er proposition og predicates?

- Uttrykk med en sannhetsverdi

p, q, r	Påstand / proposition
P(x), Q(y)	Predikat / predicate
∀x∃y P(x,y)	Quantifiers

P(x) = "The word x contains the letter **a**."

a) P(or**a**nge) = true

b) P(lemon) = false

c) P(true) = false

d) P(f**a**lse) = true

P(x) = “x can speak Russian”

Q(x) = “x knows the computer language C++”

Logikk - Definisjoner

Negation: NOT p

Conjunction: p AND q

Disjunction: p OR q

Som i programmering

p	$\neg p$
T	F
F	T

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

Logikk - Definisjoner

Implication: if p THEN q

Bijection p IFF* q

*IFF = If and only if

p	q	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

Logikk - Eksempler

Truth table

Eksamensoppgave v19

p	q	r	$\neg q$	$p \wedge \neg q$	$q \vee r$	$p \rightarrow (q \vee r)$	$(p \wedge \neg q) \rightarrow r$
T	T	T					
T	T	F					
T	F	T					
T	F	F					
F	T	T					
F	T	F					
F	F	T					
F	F	F					

Negasjon:

p	$\neg p$
T	F
F	T

p	q	r	¬q	p ∧ ¬q	q ∨ r	p → (q ∨ r)	(p ∧ ¬q) → r
T	T	T	**F**				
T	T	F	**F**				
T	F	T	**T**				
T	F	F	**T**				
F	T	T	**F**				
F	T	F	**F**				
F	F	T	**T**				
F	F	F	**T**				

Conjunction:

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

p	q	r	¬q	p ∧ ¬q	q ∨ r	p → (q ∨ r)	(p ∧ ¬q) → r
T	T	T	F	F			
T	T	F	F	F			
T	F	T	T	T			
T	F	F	T	T			
F	T	T	F	F			
F	T	F	F	F			
F	F	T	T	F			
F	F	F	T	F			

Disjunction:

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

p	q	r	$\neg q$	$p \wedge \neg q$	$q \vee r$	$p \rightarrow (q \vee r)$	$(p \wedge \neg q) \rightarrow r$
T	T	T	F	F	T		
T	T	F	F	F	T		
T	F	T	T	T	T		
T	F	F	T	T	F		
F	T	T	F	F	T		
F	T	F	F	F	T		
F	F	T	T	F	T		
F	F	F	T	F	F		

Implication:

p	q	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

p	q	r	$\neg q$	$p \wedge \neg q$	$q \vee r$	$p \rightarrow (q \vee r)$	$(p \wedge \neg q) \rightarrow r$
T	T	T	F	F	T	T	T
T	T	F	F	F	T	T	T
T	F	T	T	T	T	T	T
T	F	F	T	T	F	F	F
F	T	T	F	F	T	T	T
F	T	F	F	F	T	T	T
F	F	T	T	F	T	T	T
F	F	F	T	F	F	T	T

Logikk - Logiske ekvivalenser.

De morgans lover: Viktig!

Brukes for å skrive om uttrykk

Disse er spesielt viktig: ⇨

Equivalence	*Name*
$p \wedge \mathbf{T} \equiv p$ $p \vee \mathbf{F} \equiv p$	Identity laws
$p \vee \mathbf{T} \equiv \mathbf{T}$ $p \wedge \mathbf{F} \equiv \mathbf{F}$	Domination laws
$p \vee p \equiv p$ $p \wedge p \equiv p$	Idempotent laws
$\neg(\neg p) \equiv p$	Double negation law
$p \vee q \equiv q \vee p$ $p \wedge q \equiv q \wedge p$	Commutative laws
$(p \vee q) \vee r \equiv p \vee (q \vee r)$ $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$	Associative laws
$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$ $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$	Distributive laws
$\neg(p \wedge q) \equiv \neg p \vee \neg q$ $\neg(p \vee q) \equiv \neg p \wedge \neg q$	De Morgan's laws
$p \vee (p \wedge q) \equiv p$ $p \wedge (p \vee q) \equiv p$	Absorption laws
$p \vee \neg p \equiv \mathbf{T}$ $p \wedge \neg p \equiv \mathbf{F}$	Negation laws

Logikk - Logiske ekvivalenser.

$$p \to q \equiv \neg p \vee q$$

$$p \to q \equiv \neg q \to \neg p$$

$$p \vee q \equiv \neg p \to q$$

$$p \wedge q \equiv \neg(p \to \neg q)$$

$$\neg(p \to q) \equiv p \wedge \neg q$$

$$(p \to q) \wedge (p \to r) \equiv p \to (q \wedge r)$$

$$(p \to r) \wedge (q \to r) \equiv (p \vee q) \to r$$

$$(p \to q) \vee (p \to r) \equiv p \to (q \vee r)$$

$$(p \to r) \vee (q \to r) \equiv (p \wedge q) \to r$$

De her er veldig viktig

$$p \leftrightarrow q \equiv (p \to q) \wedge (q \to p)$$

$$p \leftrightarrow q \equiv \neg p \leftrightarrow \neg q$$

$$p \leftrightarrow q \equiv (p \wedge q) \vee (\neg p \wedge \neg q)$$

$$\neg(p \leftrightarrow q) \equiv p \leftrightarrow \neg q$$

Logikk - Logiske ekvivalenser.

Use basic logical equivalences to prove that $(p \land \neg q) \rightarrow \neg r$ and $(p \land r) \rightarrow q$ are logically equivalent.

1 $p \rightarrow q \quad = \quad \neg p \lor q$

2 $\neg(p \land q) \quad = \quad \neg p \lor \neg q$

3 $\neg(\neg p) \quad = \quad p$

$$(p \land \neg q) \rightarrow \neg r$$

$$\neg\,(p \land \neg q) \lor \neg r$$

$$(\neg p \lor q) \lor \neg r$$

$$\neg p \lor q \lor \neg r$$

$$\neg p \lor \neg r \lor q$$

$$\neg(p \land r) \lor q$$

$$(p \land r) \rightarrow q$$

1

2 3

2

1

Logikk - Quantifiers

$\forall$xP(x) - for all x in S, P(x) is true

$\exists$xP(x) - there exist x in S such that P(x) is true

Domene: Settet med elementer vi tar hensyn til

Statement	***When True?***	***When False?***
$\forall x P(x)$	$P(x)$ is true for every x.	There is an x for which $P(x)$ is false.
$\exists x P(x)$	There is an x for which $P(x)$ is true.	$P(x)$ is false for every x.

Logikk - Quantifiers

Predicate

$\forall x\, P(x) = P(x_1) \land P(x_2) \land \ldots \land P(x\square)$

$\exists x\, P(x) = P(x_1) \land P(x_2) \land \ldots \land P(x\square)$

De Morgan's Laws

$\neg \forall x\, P(x) = \exists x \neg P(x)$

$\neg \exists x\, P(x) = \forall x \neg P(x)$

P(x): x wears glasses
Q(x): x wears a hat

$\exists x\, P(x) \land \exists x\, Q(x)$ → True
$\exists x\, (P(x) \land Q(x))$ → False
$\forall x\, P(x) \lor \forall x\, Q(x)$ → False
$\forall x\, (P(x) \lor Q(x))$ → True

Logikk - Nested quantifiers

Nested Quantifiers - flere quantifiers i samme setning

$\forall x \ \exists y$ (P(x, y))

$\exists y \forall x$(P(x) -> Q(y))

Eksempel eksamen v19:

Let $\mathrm{P(n,\ m)}$ be “n is greater than or equal to m” where to domain if the set of nonnegative integers. What are the truth values for $\exists \mathrm{n} \forall \mathrm{m P(n,m)}$ and $\forall \mathrm{n} \exists \mathrm{m P(n,m)}$?

1. Kan du velge en *n* slik at den verdien er større eller lik alle *m*
2. For alle *n*, kan du velge en *m* slik at denne er større eller lik *n*

Logikk - Nested quantifiers

Hva er forskjellen på:

$\forall x \exists y(x + y = 0)$	1. For alle x finnes det en y slik at x + y = 0 er sant
$\exists y \forall x(x + y = 0)$	2. Det finnes en y slik at for all x, x + y = 0 er sant

1. Svar: For alle x kan du velge y til å være -x
2. Svar: Det finnes ingen y slik at uansett hvilket tall x du adderer det med blir summen 0

Bevis - Viktige utgangspunkt

- Partall kan skrives på formen $2n$ for some n
- Oddetall kan skrives på formen $2n + 1$ for some n
- $a \mid b$: $b/a = c \rightarrow b = ac$

Bevis - Types of Proof

Direct Proof : n = odd → n² = odd

Vi vet at n er oddetall, og kan dermed skrives som $2m + 1 = n$

Vi tar så denne verdien og opphøyer i andre

$(2m + 1)^2$

$= 4m^2 + 4m + 1$

$= 2(2m^2 + 2m) + 1 = 2k + 1$

Proof by contraposition:

Ønsker å vise: n^2 = partall → n = partall

Kontrapositive: $p \rightarrow q \equiv \neg q \rightarrow \neg p$

p = n² er partall, q = n er partall

¬p = n² er odd, ¬q = n er odd

n = odd → n² = odd

Forenkling av brøk $\frac{16}{6} = \frac{2 \cdot 8}{2 \cdot 3} = \frac{8}{3}$

Bevis - Types of Proof

Proof by contradiction: $\sqrt{2}$ er irrasjonalt

Anta at vi kan skrive $\sqrt{2}$ som a / b, hvor a,b $\in \mathbb{Z}$ (*definisjon av rasjonale tall*) og a / b er på forenklet form (*a og b kan ikke begge være partall*)

$\sqrt{2} = a / b \rightarrow 2 = a^2 / b^2 \rightarrow a^2 = 2 \cdot b^2$

Fra beviset fra forrige side så vet vi om a^2 er partall, så er a partall, a = 2k

$2 = (2k)^2 / b^2 \rightarrow 2 = 4k^2 / b^2 \rightarrow 2b^2 = 4k^2 \rightarrow b^2 = 2k^2$

Contradiction: a er partall **og** b er partall

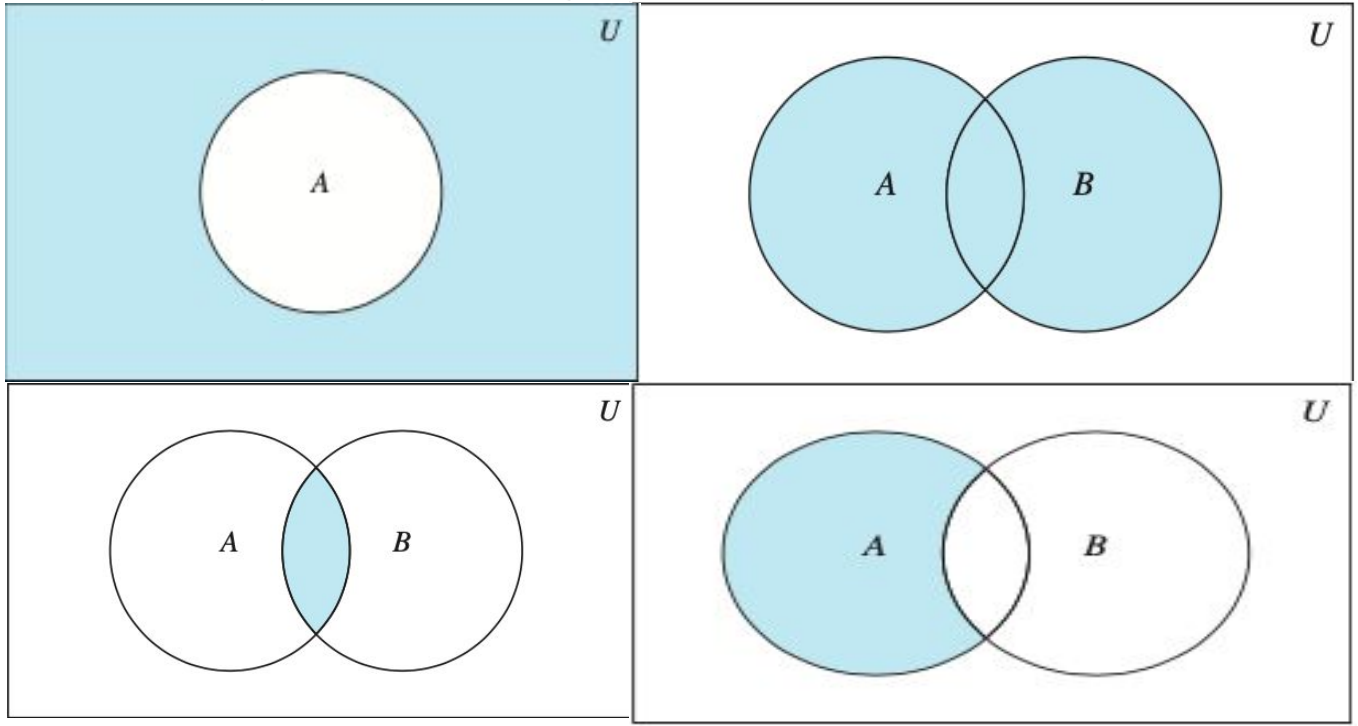
Mengdelære / Set Theory - Definisjoner

Definisjon av set

- $S = \{a, b, c\}$: et sett med elementer a, b, og c
- $S \{a, a, a, b, b, c, \} = \{a, b, c\}$, ikke lov med repeterte element.
- Null settet Ø. {}
- Kardinalitet: Antall elementer i settet.
- Set builder notation: $\{x \in S : P(x)\}$, x er medlem i settet hvis P(x) = T
- Powerset P(S) : $\{x: x \subseteq S\}$
- Cartesian Product $A \times B = \{(a,b) : a \in A, b \in B \}$

Mengdelære / Set Theory - Operasjoner

- Complement
- Union
- Snitt
- Difference



Set Operasjoner - Set builder notation

$$\bar{A} = \{x | x \notin A\}$$

$$A \cup B = \{x | x \in A \vee x \in B\}$$

$$A \cap B = \{x | x \in A \wedge x \in B\}$$

$$A - B = \{x | x \in A \wedge x \notin B\}$$

Mengdelære / Set Theory - Set identiteter

Merk sammenhengen med logiske ekvivalenser fra kapittel 1.

$$A \cup (B \cap C) = \{x | x \in A \vee x \in (B \cap C)\}$$
$$= \{x | x \in A \vee x(x \in B \wedge x \in C)\}$$
$$= \{x | (x \in A \vee x \in B) \wedge (x \in A \vee x \in B)\}$$
$$= (A \cup B) \cap (A \cup C)$$

Identity	*Name*
$A \cap U = A$ $A \cup \emptyset = A$	Identity laws
$A \cup U = U$ $A \cap \emptyset = \emptyset$	Domination laws
$A \cup A = A$ $A \cap A = A$	Idempotent laws
$\overline{(\overline{A})} = A$	Complementation law
$A \cup B = B \cup A$ $A \cap B = B \cap A$	Commutative laws
$A \cup (B \cup C) = (A \cup B) \cup C$ $A \cap (B \cap C) = (A \cap B) \cap C$	Associative laws
$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$	Distributive laws
$\overline{A \cap B} = \overline{A} \cup \overline{B}$ $\overline{A \cup B} = \overline{A} \cap \overline{B}$	De Morgan's laws
$A \cup (A \cap B) = A$ $A \cap (A \cup B) = A$	Absorption laws
$A \cup \overline{A} = U$ $A \cap \overline{A} = \emptyset$	Complement laws

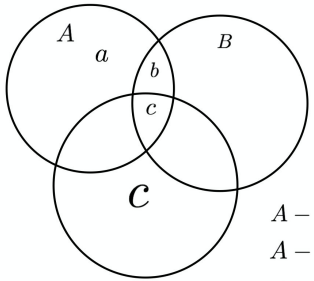
Mengdelære / Set Theory - Eksempel:

Eksamen v19

a) Let A, B, C be sets. Prove or disprove (give a counterexample) that $A \setminus (B \cap C) = (A \setminus B) \cap (A \setminus C)$. (Note: "$\setminus$" denotes the set difference operator, sometimes also denoted "$-$")

b) Let A, B, C be sets. Prove or disprove that $A \setminus (B \cap C) = (A \setminus B) \cup (A \setminus C)$.

Set Theory - Eksempel.



$A = \{a, b, c\}$
$B = \{b, c\}$
$C = \{c\}$

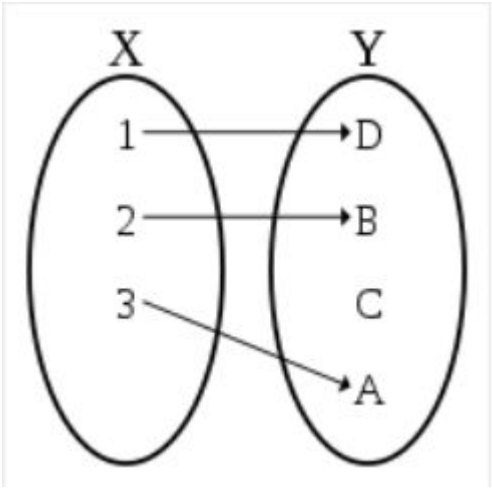
$A - (B \cup C) = \{a, b\}$
$A - B = \{a\}$
$A - C = \{a, b\}$
$(A - B) \cap (A - C) = \{a\}$

$$
\begin{aligned}
A - (B \cap C) &= A \cap \overline{(B \cap C)} \\
&= A \cap (\bar{B} \cup \bar{C}) \\
&= A \cap \bar{B} \cup A \cap \bar{C} \\
&= (A - B) \cup (A - C)
\end{aligned}
$$

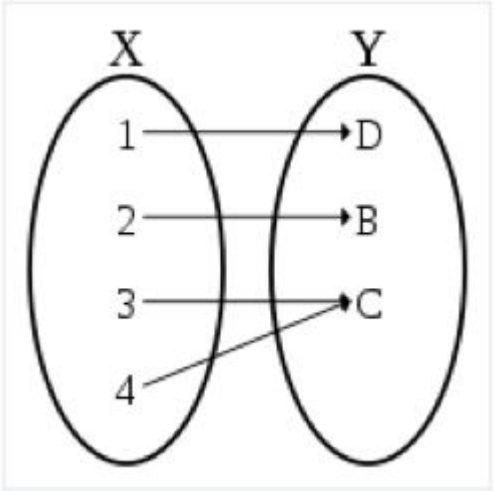
Set Theory - Funksjoner

- Domene og Co-domene (aka Definisjonsmengde og Verdimengde)
- Notasjon: $f : A \to B$
  - der $f \subseteq A \times B$
- En funksjon er
  - Injektiv når: $\forall_{a \in A} \forall_{b \in B} f(a) = f(b) \to a = b$
    - To verdier i domene går ikke til samme verdi i Co-domene.
  - Surjektiv når: $\forall_{b \in B} \exists_{a \in A} b = f(a)$
    - Alle verdier i Co-domene har en verdi i Domene
  - Bijektiv når Injektiv og surjektiv.

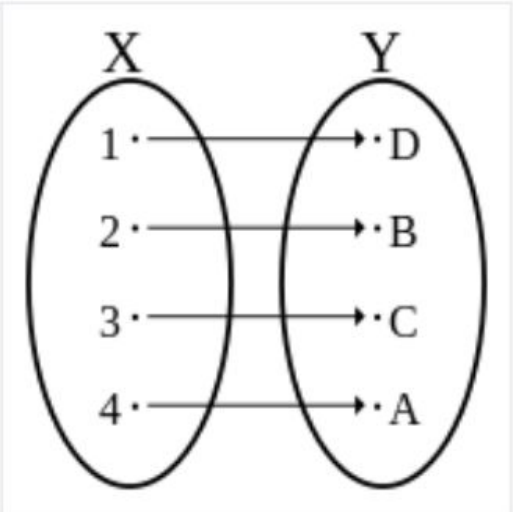
Set Theory - Funksjoner



Injektiv



Surjektiv



Bijektiv

Set Theory - Funksjoner

Eksempel Eksamen v19

c) Let $f(n) = 3n$ be a function from the set of integers to the set of integers. Is f one-to-one (injective)? Onto (surjective)? A one-to-one correspondence (bijective)? Explain why/why not.

Funksjoner eksempel

$$f : \mathbb{Z} \to \mathbb{Z}$$
$$f(n) = 3n$$

Funksjoner - Injektiv?

$$\text{la } a, a' \in \mathbb{Z}$$
$$f(a) = 3a$$
$$f(a') = 3a'$$
$$\text{hvis } a = a' \rightarrow f(a') = f(a) = 3a$$
$$\text{hvis } a \neq a' \rightarrow f(a) \neq f(a') \rightarrow 3a \neq 3a'$$

Injektiv ✓

Funksjoner - Surjektiv?

Ikke alle tall er i 3-gangen.

$$f(n) = 1$$
$$3n = 1$$
$$n = \frac{1}{3} \notin \mathbb{Z}$$

Surjektiv ❌

Funksjoner Eksempel.

$$f : \mathbb{Z} \to \mathbb{Z}$$
$$f(n) = 3n$$

Injektiv ✓

Surjektiv ❌

Bijektiv ❌

Sekvenser - Recurrence relations:

- Recurrence relation:
  - starter med et element a_0
  - a_1 er definert ut ifra a_0
  - a_n = definert ut ifra a_{n-1}

$$a_0 = 5$$
$$a_n = a_{n-1} + 4$$
$$a_1 = 5 + 4 = 9$$
$$a_2 = 9 + 4 = 13$$
$$\dots$$

Fibonacci tall:

$$f_0 = 0, f_1 = 1$$
$$f_n = f_{n-2} + f_{n-1}$$

- Veldig nyttig i koding:
  - Eksempel: fakultet tall

$$f_0 = 1$$
$$f_n = n \cdot f_{n-1}$$

```
def factorial(n): #
    if n == 0: return 1 # f_0
    else:
        return n * factorial(n-1) # f_{n-1} * n
```

Rekker

Summetegnet sigma.

$$\sum_{i=0}^{n} i = 1 + 2 + \ldots + n$$

Dobbel summasjon:

$$\begin{aligned}\sum_{i=1}^{4}\sum_{j=1}^{3} ij &= \sum_{i=1}^{4}(i + 2i + 3i) \\ &= \sum_{i=1}^{4} 6i \\ &= 6 + 12 + 18 + 24 = 60.\end{aligned}$$

Rekker - Geometrisk rekke

- Geometrisk rekke har formen a*r^i
  - a er initial leddet og r er “common ratio”
  - Eksempel : $\{2, 6, 18, 54...\} : 2 \cdot 3^i, (a = 2, r = 3)$
- Summen av en geometrisk rekke:
  - Eksempel:

$$\sum_{i=0}^{n} ar^i = \begin{cases} \dfrac{ar^{n+1} - a}{r - 1}, r \neq 1 \\ (n+1)\, a, r = 1 \end{cases}$$

$$\sum_{i=0}^{3} 2 \cdot 3^i = \frac{2 \cdot 3^{3+1} - 2}{3 - 1} = \frac{2 \cdot 3^4 - 2}{2} = \frac{160}{2} = 80$$

$$\begin{aligned} \sum_{i=0}^{3} 2 \cdot 3^i &= 2 \cdot 3^0 + 2 \cdot 3^1 + 2 \cdot 3^2 + 2 \cdot 3^3 \\ &= 2 + 6 + 18 + 54 \\ &= 80 \end{aligned}$$

Set - Kardinalitet

- Husk: |A| = antall elementer i A.
- A = B iff A ⊆ B and B ⊆ A
- Countability:
  - Et set kan være countably infinite, da blir kardinaliteten aleph 0

$$|\mathbb{N}| = |\mathbb{Z}| = |\mathbb{Q}| = \aleph_0$$

  - Et set kan være uncountable

$$|\mathbb{N}| < |\mathbb{R}|$$

Tallteori - Division og Mod

- a|b: a deler b.
  - Husk: a | b → b / a = c → b = a · c, der c er et heltall
  - 3 | 12 → 12 / 3 = 4 → 12 = 3 · 4
- mod
  - “Clock Arithmetic”
  - a mod b gir det som står igjen etter første steg i lang divisjon.
  - a mod b = r ← remainder
  - 10 mod 3 = 1 siden 10 = 3 * 3 + 1 ← remainder
  - Division algorithm:
    - d = qa + r
      - der q = d div a
      - og r = d mod a
      - i eksempelet ovenfor er a = 10 div 3 (floor(10 / 3)) og r = 10 mod 3

Tallteori - Mer mod

- Kongruenser
  - $a \equiv b (\mathrm{mod}\ m) : a \bmod m = b \bmod m$
- regneregler
  - $(a + b) \mathrm{mod}\ m = (a \bmod\ + b \bmod m) \bmod m$
  - $(ab) \bmod m = (a \bmod m)(b \bmod m) \bmod m$
  - Mod likninger fungerer som vanlige likninger når det gjelder addisjon:
    - $a \bmod m = b \bmod m \rightarrow a + 1 \bmod m = b + 1 \bmod m$

Oppgave

Eksamen V19

Let $x \equiv 3 \pmod 5$ and $y \equiv 4 \pmod 5$.

- What is the value of $(57x^3) \bmod 5$?
- What is the value of $(3x + 2y^2) \bmod 5$?

Oppgave - Løsning

$$(57x^3) \bmod 5 = (57 \bmod 5)(x \bmod 5)(x \bmod 5)(x \bmod 5) \bmod 5$$

$$x \equiv 3(\bmod 5) \rightarrow x = 3$$

$$(57 \cdot x^3) = (57 \bmod 5)(3 \bmod 5)(3 \bmod 5)(3 \bmod 5) \bmod 5$$

$$(2 \cdot 3 \cdot 3 \cdot 3) \bmod 5$$

$$54 \bmod 5 = 4$$

$$(3x + 2y^2) \bmod 5 = ((3x) \bmod 5 + (2y^2) \bmod 5) \bmod 5$$

$$(3x) \bmod 5 = (3 \bmod 5)(x \bmod 5) \bmod 5$$

$$x \equiv 3(\bmod 5) \rightarrow x = 3$$

$$= (3 \bmod 5)(3 \bmod 5) \bmod 5 = 4$$

$$9 \bmod 5 = 4$$

$$(3x) \bmod 5 = 4$$

$$(2y^2) \bmod 5 = (2 \bmod 5)(y \bmod 5)(y \bmod 5) \bmod 5$$

$$y \equiv 4(\bmod 5) \rightarrow y = 4$$

$$(2y^2) \bmod 5 = (2 \bmod 5)(4 \bmod 5)(4 \bmod 5) \bmod 5$$

$$= 32 \bmod 5 = 2$$

$$(2y^2) \bmod 5 = 2$$

$$(3x) \bmod 5 = 4$$

$$(2y^2) \bmod 5 = 2$$

$$(3x + 2y^2) \bmod 5$$

$$= (4 + 2) \bmod 5$$

$$= 6 \bmod 5 = 1$$

Tallteori - Tallsystemer / Baser

- Desimal “base 10”
  - Teller med ti siffer 0-9
- Binær “base 2”
  - Teller med 2 siffer 0-1
- Octal “base 8”
  - Teller med siffer 0-7
- Hexadecimal “base 16”
  - Teller med 16 siffer 0-9 og A-F <- bokstavene a-f er siffer i base 16

Tallteori - Tallsystemer / baser

- Konvertering
  - mod og div
    - n mod 2 for å finne siffer i binær ekspansjon
    - n div 2 for å gå til neste steg
  - 120 = 1111000 (base 2)
  - 120 = 120 (base 8)

$$
\begin{array}{ll}
120 \bmod 2 = 0 & 120 \operatorname{div} 2 = 60 \\
60 \bmod 2 = 0 & 60 \operatorname{div} 2 = 30 \\
30 \bmod 2 = 0 & 30 \operatorname{div} 2 = 15 \\
15 \bmod 2 = 1 & 7 \operatorname{div} 2 = 7 \\
7 \bmod 2 = 1 & 3 \operatorname{div} 2 = 3 \\
3 \bmod 2 = 1 & 1 \operatorname{div} 2 = 1 \\
1 \bmod 2 = 1 & 0 \operatorname{div} 2 = 0
\end{array}
$$

```
In [8]: def bin_expansion(n):
   ...:     while n != 0:
   ...:         print(n, n % 2)
   ...:         n = n // 2
   ...:
```

```
In [14]: def base_b_expand(n, b):
    ...:     while n != 0:
    ...:         print(n, n % b)
    ...:         n = n // b
    ...:
```

Denne funksjonen fungerer ikke for b > 10

Tallteori - Primtall, gcd, og lcm

- Primtall: tall som har bare 1 og seg selv som faktor.
- gcd(a,b): greatest common divisor (største felles faktor)
  - To tall a,b er “Relativt primisk” hvis gcd(a,b) = 1
  - aka “co-prime”
- lcm(a,b): least common multiple (det minste tallet som er delelig på a og b)
  - hvis gcd(a,b) = 1 -> lcm(a,b) = ab
- Theorem:
- Let a and b be positive integers. Then

  $$ab = \gcd(a, b) \cdot \operatorname{lcm}(a, b).$$

Tallteori - Euklids algoritme:

- Eksempel: gcd(22, 13)
- $gcd(22, 13)$

$22 = (1)13 + 9$ ⇐ 22 = (22 div 13) * 13 + (22 mod 13)

$13 = (1)9 + 4$

$9 = (2)4 + 1$ ⇐ gcd(22, 13)

$4 = (4)1 + 0$

Tallteori - Extended euclidean algorithm

- Bezout koeffisienter: skrive gcd(a, b) som en lineær kombinasjon av a og b
  - altså: gcd(a,b) = sa + tb
- Brukes til å finne modulære inverser
  - Merk: modulære inverser finns iff gcd(a,b) = 1 (a og b er relatively prime)
  - eksempel (fra forrige slide)

Note: Modulære inverser finnes iff gcd(a,b) = 1

$$\text{eksemepel fra forige slide}$$

$$\text{vi vet at } gcd(22, 13) = 1$$

$$22 = 1(13) + 9$$

$$13 = (1)9 + 4$$

$$9 = (2)4 + 1$$

$$1 = 9 + (-2)4 \quad \Leftarrow \quad 4 = 13 + (-1)9$$

$$1 = 9 + (-2)(13 + (-1)9) \quad \Leftarrow \quad 9 = 22 + (-1)13$$

$$1 = 22 + (-1)13 + (-2)13 + (2)22 + (-2)13$$

$$1 = (2)22 + (-5)13$$

$$-5 \bmod 22 = 17 \bmod 22$$

Kryptografi

- Symmetrisk kryptografi (aka klassisk kryptografi)
  - Krever at begge parter har en nøkkel for kryptering og dekryptering
  - Caesar cipher f(c) = (c + key) mod 26 (key er offset på alfabetet)
- Asymmetrisk kryptografi:
  - Privat og offentlig nøkkel
  - RSA

Kryptografi - RSA

- Asymmetrisk
- Alle har en offentlig nøkkel og en privat nøkkel.
- I kryptering:
  - Offentlig nøkkel blir brukt til kryptering
  - Privat nøkkel blir brukt til dekryptering
  -
- I Digital signature:
  - Privat nøkkel blir brukt til å signere
  - Offentlig nøkkel blir brukt til å verifisere
- Dette kurset fokuserer på kryptering

Kryptografi - RSA : kryptering

- Velg 2 primtall p og q
- Regn ut $n = pq$
- Regn så ut: $\phi(n) = (p-1)(q-1)$
- Velg så en e, s.a $2 < e < \phi(n), gcd(e, \phi(n)) = 1$
  - Dette brukes for å finne de-krypteringsnøkkelen.
- Bruk EEA for å finne $d = e^{-1} \mod \phi(n)$
- Setup ferdig.
- Viktig å dele meldingen inn i blocks.
- Krypter hver blokk M med $C = M^e \mod n$
- Decrypter hver blokk C med $M = C^d \mod n$
  - C^d mod n = M^{e*e^-1} mod n

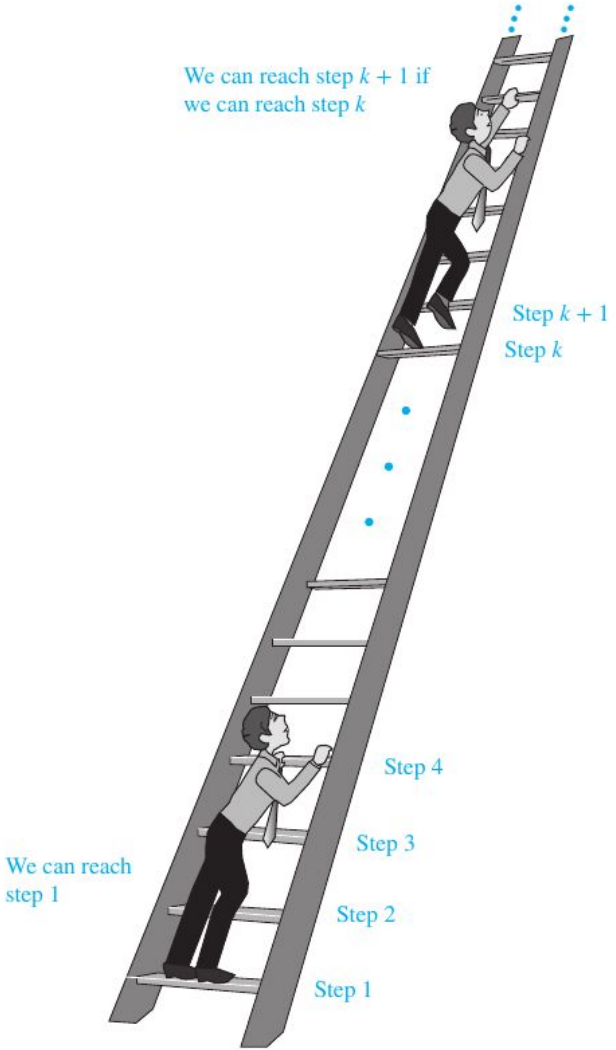
Induksjon

Mathematical induction

Strong induction

Structural induction

- Recursion



Induksjon - Mathematical induction

Induksjon i 3 deler:

Basis steg

For all integers n ≥ b

Sjekke at P(b) er sann

Induksjonshypotese

Anta at P(k) er sann, for en vilkårlig k ≥ b

Induksjonssteget

Vise at om P(k) er sann er P(k+1) sann

Hvordan ser problemet ut?

Use mathematical induction to *prove that*

$$\sum_{k=0}^{n} 3^k = \frac{1}{2}\left(3^{n+1} - 1\right)$$

for all integers $n \geq 0$.

Use mathematical induction to *prove that*

$$\sum_{k=1}^{n} k^2 = \frac{n(n+1)(n+2)}{6}$$

for all integers $n \geq 1$.

Induksjon - Mathematical induction - Eksempel

$$\text{Define } P(n)$$

$$P(n) = \left(\sum_{k=0}^{n} 3^k = \frac{1}{2}\left(3^{n+1} - 1\right)\right)$$

$$\text{Basis step} - \text{show } P(0)$$

$$\sum_{k=0}^{0} 3^k = \frac{1}{2}\left(3^1 - 1\right)$$

$$3^0 = 1 = \frac{1}{2} \cdot 2 = 1$$

$$\text{Hence } P(n) \text{ is true for } n = 0$$

$$\text{Use mathematical induction to prove that}$$

$$\sum_{k=0}^{n} 3^k = \frac{1}{2}\left(3^{n+1} - 1\right)$$

$$\text{for all integers } n \geq 0.$$

$$\text{Induction hypothesis}$$

$$\text{Assume } P(m) \text{ is true}$$

$$\text{for some arbitrary } m \geq 0$$

Induksjon - Mathematical induction - Eksempel

Use mathematical induction to *prove that*

$$\sum_{k=0}^{n} 3^k = \frac{1}{2}\left(3^{n+1}-1\right)$$

for all integers $n \geq 0$.

Induction hypothesis

$$\sum_{k=0}^{m} 3^k = \frac{\left(3^{m+1}-1\right)}{2}$$

Induction step – verify $P(m+1)$

$$\sum_{k=0}^{m+1} 3^k = \left(\sum_{k=0}^{m} 3^k\right) + 3^{m+1}$$

$$= \frac{3^{m+1}-1}{2} + 3^{m+1}$$

$$= \frac{3^{m+1} + 2\cdot 3^{m+1} - 1}{2}$$

$$= \frac{3\cdot 3^{m+1} - 1}{2}$$

$$= \frac{3^{m+2}-1}{2}$$

Hence $P(m+1)$ *is true*

By the principle of mathematical induction,
$P(n)$ *is true for all* $n \geq 0$.

Induksjon - Strong induction

Induksjon i 3 deler:

Basis steg

For all integers n ≥ b

Sjekke at P(b) er sann

Induksjonshypotese

Anta at P(b), P(b+1), P(b+2)... P(k) er sann

eller anta at P(i), b ≦ i ≦ k er sann

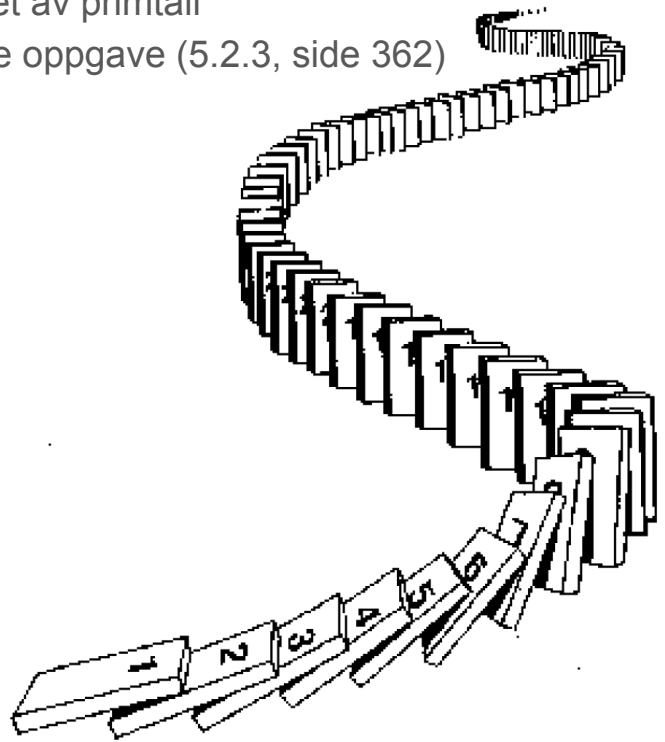
for en vilkårlig verdi k

Induksjonssteget

Vise at P(k+1) er sann om alle verdiene opptil P(k) er sann

Eksempler:

1. Vise at alle tall >= 2 kan skrives som produktet av primtall
2. Frimerke oppgave (5.2.3, side 362)



Induksjon - Strong induction - Primtallsfaktorisering

Vise at alle tall >= 2 kan skrives som produktet av primtall

P(n): n har en primtallsfaktorisering

Basis steg: P(2) er sann fordi 2 = 2, og 2 er et primtall

Induksjonshypotese: anta at alle tall opptil k kan skrives som et produkt av primtall

Induksjonssteget: For P(k+1) så har vi to cases:

1. k+1 er et primtall -> P(k+1) er sann
2. om k+1 ikke er et primtall finnes det finnes to tall p, q = k+1. Siden p, q < k så vil disse to verdiene ha en primtallsfaktorisering, og k+1 er produktet av disse to.

Induksjon i 3 deler:

Basis steg

For all integers $n \geq 2$

Sjekke at P(2) er sann

Induksjonshypotese

Anta at P(2), P(3), P(4) ... P(k) er sann

eller anta at P(i), $2 \leqq i \leqq k$ er sann

for en vilkårlig verdi k

Induksjonssteget

Vise at P(k+1) er sann om alle verdiene opptil P(k) er sann

Induksjon - Strong induction - Frimerker

Let P(n) be the statement that postage of n cents can be formed using 3-cent and 5-cent stamps. Show that is is true for all integers n >= 8

Basis steg:

P(8) = 5 + 3 = 8

P(9) = 3 + 3 + 3 = 9

P(10) = 5 + 5

Legge merke til at P(11) = P(8) + 3

P(12) = P(9) + 3 …

Så vi tar å beviser 8, 9, 10

Induksjon i 3 deler:

Basis steg

For all integers $n \geq 8$

Sjekke at P(8), P(9), P(10) er sann

Induksjonshypotese

Anta at P(8), P(9), P(10), P(11) ... P(k) er sann

eller anta at P(i), $8 \leqq i \leqq k$ er sann

for en vilkårlig verdi k

Induksjonssteget

Vise at P(k+1) er sann om alle verdiene opptil P(k) er sann

Induksjon - Strong induction - Frimerker

Let P(n) be the statement that postage of n cents can be formed using 3-cent and 5-cent stamps. Show that is is true for all integers n >= 8

Induksjonshypotese:

Anta at P(8), P(9) .... P(k) er sann

Induksjonssteget:

Den minste verdien vi kan velge for k slik at k+1 er ukjent er k = 10

Fra induksjonshypotesen vet vi at P(k-2) er sann.

Om k = 10, får vi P(10-2) = P(8) som er sann

For å vise at P(k+1) er sann, bruker vi at P(k-2) + 3 = P(k+1)

Induksjon i 3 deler:

Basis steg

For all integers n ≥ 8

Sjekke at P(8), P(9), P(10) er sann

Induksjonshypotese

Anta at P(8), P(9), P(10), P(11) ... P(k) er sann

eller anta at P(i), $8 \leqq i \leqq k$ er sann

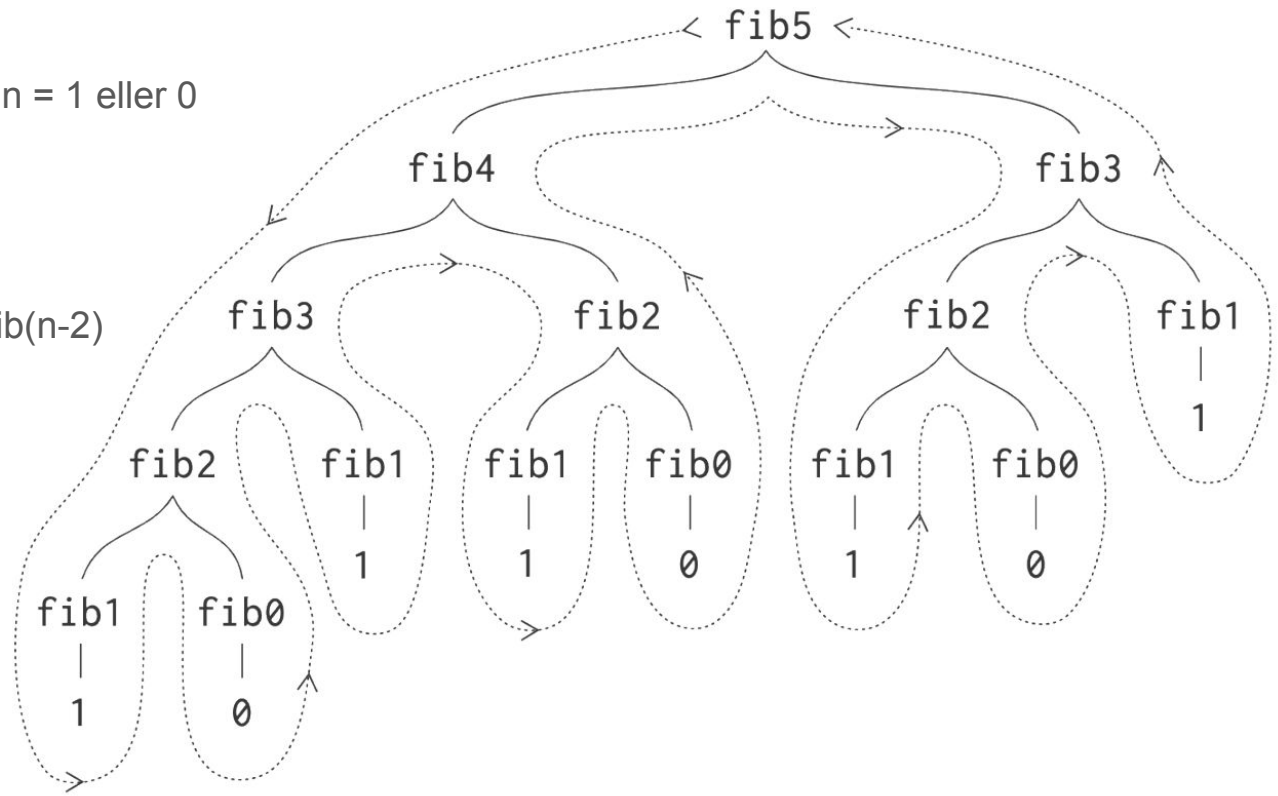
for en vilkårlig verdi k

Induksjonssteget

Vise at P(k+1) er sann om alle verdiene opptil P(k) er sann

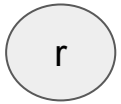
Rekursjon

```
fib(n):
    if (n < 2):          # om n = 1 eller 0
        return 1
    else:
        return fib(n-1) + fib(n-2)
```



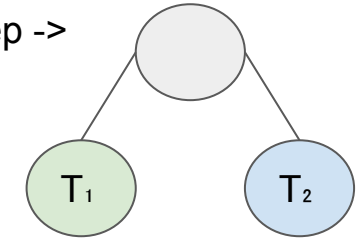
Rekursjon - Trees

Basis step ->



Definisjon av Full Binary Trees:

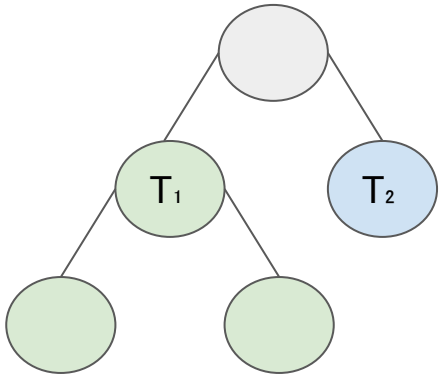
1. Recursive step ->

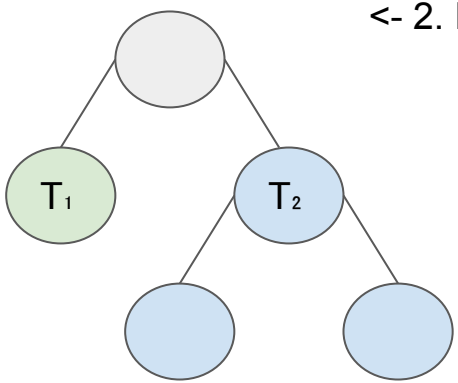


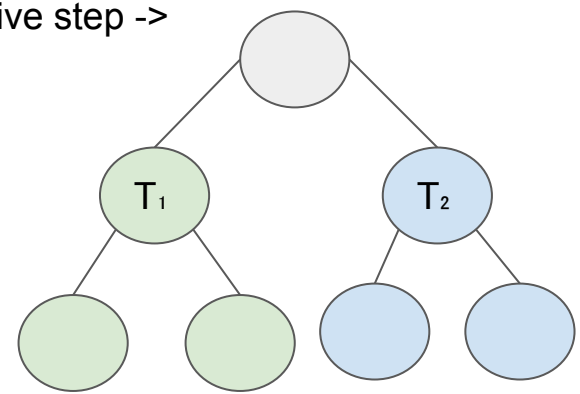
• **Basis step**: There is a full binary tree consisting of a single vertex r.

• **Recursive step**: If T_1 and T_2 are disjoint full binary trees, then the tree $T = T_1 \cdot T_2$ formed by connecting a new root r to each of the roots of the left subtree T_1 and right subtree T_2 is also a full binary tree.

<- 2. Recursive step ->

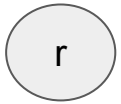






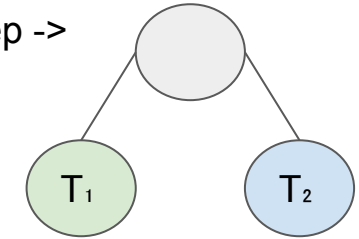
Rekursjon - Trees

Basis step ->



Definisjon av antall noder $n(T)$, antall blader $l(T)$

1. Recursive step ->



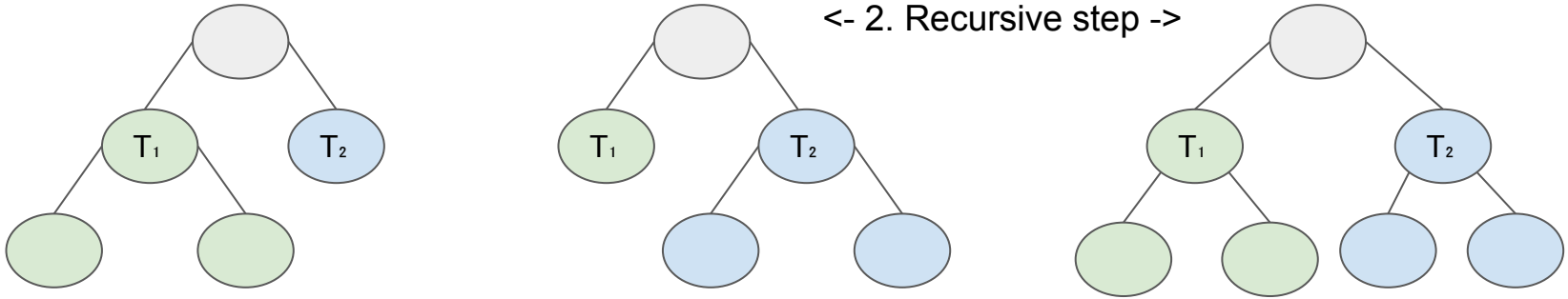
• **Basis step**: T is a full binary tree consisting of a single vertex, then $n(T) = 1$, $l(T) = 1$

• **Recursive step**: If T_1 and T_2 are disjoint full binary trees, then the tree $T = T_1 \cdot T_2$ formed by connecting a new root r to each of the roots of the left subtree T_1 and right subtree T_2, then:

$$n(T) = n(T_1) + n(T_2) + 1$$

$$l(T) = l(T_1) + l(T_2)$$

<- 2. Recursive step ->



Induksjon - Structural induction

Definisjoner fra forrige slide

En node: $n(T) = 1, l(T) = 1$

$$n(T) = n(T_1) + n(T_2) + 1$$

$$l(T) = l(T_1) + l(T_2)$$

Bruker structural induction for å bevise egenskaper til rekursivt definerte sett.

Eksempel: egenskaper til rekursivt definerte “full binary tree”

Ønsker å bevise: $n(T) = 2 \cdot l(T) - 1$ for alle full binary trees

Som andre induksjonsbevis har vi tre deler:

Basis steg, Induksjonshypotese og Induksjonssteget.

Basis steg: Sjekker om det stemmer for bare 1 node

$$n(T) = 1 = 2 \cdot l(T) - 1 = 2 \cdot 1 - 1 = 1$$

Induksjonshypotesen: Anta at for to vilkårlige disjunkte full binary trees T_1 og T_2, følgende er sant:

$$n(T_1) = 2 \cdot l(T_1) - 1 \quad \text{og} \quad n(T_2) = 2 \cdot l(T_2) - 1$$

Induksjonssteget:

neste slide ->

Induksjon - Structural induction

Definisjoner fra forrige slide

$$n(T) = n(T_1) + n(T_2) + 1$$

$$l(T) = l(T_1) + l(T_2)$$

Basis steg: Sjekker om det stemmer for bare 1 node

$$n(T) = 1 = 2 \cdot l(T) - 1 = 2 \cdot 1 - 1 = 1$$

Induksjonshypotesen: Anta at for to vilkårlige disjunkte full binary trees T_1 og T_2, følgende er sant:

$$n(T_1) = 2 \cdot l(T_1) - 1 \qquad \text{og} \qquad n(T_2) = 2 \cdot l(T_2) - 1$$

Induksjonssteget:

$T = T_1 \cdot T_2$ er et fullt binary tre skapt med å koble en ny root r til røttene av T_1 og T_2

Fra den rekursive definisjonen av $n(T)$ har vi:

$$n(T) = n(T_1) + n(T_2) + 1$$

Fra induksjonshypotesen kan vi skrive $n(T_a)$:

$$n(T_a) = 2 \cdot l(T_a) - 1$$

$$n(T) = (2 \cdot l(T_1) - 1) + (2 \cdot l(T_2) - 1) + 1$$

$$n(T) = 2 \cdot (l(T_1) + l(T_2)) - 1$$

Fra den rekursive definisjonen av $l(T)$:

$$n(T) = 2 \cdot l(T) - 1$$

Counting

- Product rule
  - for hver task i n_1 kan du gjøre n_2 tasks
  - $n_1 * n_2$ ulike måter å gjøre task på
  - eksempel 6.3.29

THE PRODUCT RULE Suppose that a procedure can be broken down into a sequence of two tasks. If there are n_1 ways to do the first task and for each of these ways of doing the first task, there are n_2 ways to do the second task, then there are $n_1 n_2$ ways to do the procedure.

Counting

- Sum rule
  - Når du kan gjøre a eller b
  - THE SUM RULE If a task can be done either in one of n_1 ways or in one of n_2 ways, where none of the set of n_1 ways is the same as any of the set of n_2 ways, then there are $n_1 + n_2$ ways to do the task.

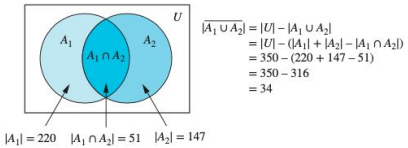
Counting

- Subtraction rule
  - Don’t count twice

$$|A_1 \cup A_2| = |A_1| + |A_2| - |A_1 \cap A_2|.$$

EXAMPLE 19 A computer company receives 350 applications from college graduates for a job planning a line of new web servers. Suppose that 220 of these applicants majored in computer science, 147

6 / Counting



$$\begin{aligned}\overline{|A_1 \cup A_2|} &= |U| - |A_1 \cup A_2| \\ &= |U| - (|A_1| + |A_2| - |A_1 \cap A_2|) \\ &= 350 - (220 + 147 - 51) \\ &= 350 - 316 \\ &= 34\end{aligned}$$

FIGURE 3 **Applicants who majored in neither computer science nor business.**

majored in business, and 51 majored both in computer science and in business. How many of these applicants majored neither in computer science nor in business?

THE SUBTRACTION RULE If a task can be done in either n_1 ways or n_2 ways, then the number of ways to do the task is $n_1 + n_2$ minus the number of ways to do the task that are common to the two different ways.

Counting

THE DIVISION RULE There are n/d ways to do a task if it can be done using a procedure that can be carried out in n ways, and for every way w, exactly d of the n ways correspond to way w.

Counting - Oppgave

24. How many positive integers between 1000 and 9999 inclusive

a) are divisible by 9?
b) are even?
c) have distinct digits?
d) are not divisible by 3?
e) are divisible by 5 or 7?
f) are not divisible by either 5 or 7?
g) are divisible by 5 but not by 7?
h) are divisible by 5 and 7?

a) først finne antall tall delelig med 9, fra 0 til 9999. Så finne antall tall delelig med 9 fra 0 til 1000, og trekke disse fra totalen.

d) løs a) for 3, trekk fra totalen

h) finn alle tall delelig med $(5 \cdot 7)$

e) løs a) for 5 + a) for 7, men her vil vi telle dobbelt for alle tall som er delelig med både 5 og 7, så trekk fra h)

g) løs a) for 5 men trekk fra h)

Counting - Permutations and Combinations

- Permutasjoner P(n, r)
  - Ordnet utvalg.
  - 123 og 321 telles som forskjellig

$$P(n,r) = \frac{n!}{(n-r)!}$$

- Combinations C(n, r)
  - Uordnet utvalg.
  - 123 og 321 telles som samme

$$C(n,r) = \frac{n!}{r!(n-r)!}$$

Counting - Eksempel

Thirteen people on a softball team show up for a game.

a) How many ways are there to choose 10 players to take the field?

- Her har du et utvalg av 13 personer, du skal velge 10 av disse, dette er uten tilbakelegging siden når en person er valgt kan den ikke bli valgt på nytt.

  Hvilken rekkefølge vi velger personer spiller ikke rolle, siden vi er bare interessert i settet av personer på laget.

  - C(13, 10) = 13!/(10!*3!) = 11*12*13/1*2*3 = 11*13*2 = 286

b) How many ways are there to assign the 10 positions by selecting players from the 13 people who show up?

- Her skal vi se på hvor mange permutasjoner vi kan lage når vi har 10 plasser og 13 elementer.

  Da kan vi bruke formelen P(13, 10) = 13!/(13-10)! = 13!/3!

Counting - Eksempel

c) Of the 13 people who show up, three are women. How many ways are there to choose 10 players to take the field if at least one of these players must be a woman?

Fra oppgave a) vet vi at vi totalt kan lage 286 forskjellige lag.

Denne oppgaven har to fremgangsmåter.

1) Finne ut hvor mange lag som har ingen kvinner, og trekke fra totalen.

- Siden vi har 10 menn og skal lage et lag med 10 spiller får vi C(10,10) = 1
-
    - Dette gir 286-1 = 285

2) Finne antall lag kombinasjoner med 1K/9M, 2K/8M, 3K/7M

- 1K/9M, her vil vi se på hvor mange måter kan vi velge 1K og 9M
    - 1K, her får vi C(3, 1) siden vi skal velge 1 av 3, og C(10,9). Siden hvor hver av måtene vi velger K, så kan vi velge C(10,9) M.

  Derfor må vi bruke multiplikasjonsregelen. C(3,1)*C(10,9)

- Så gjør vi det samme for de to andre, C(3,2)*C(10,8) og C(3,3)*C(10,7)

Siden lagene vi lager kan ENTEN ha 1 kvinne, 2 kvinner, eller 3 kvinner må vi bruke addisjonsregelen

C(3,1)*C(10,9) + C(3,2)*C(10,8) + C(3,3)*(10,7)

3 * 10 + 3 * 45 + 1 * 120

30 + 135 + 120

285

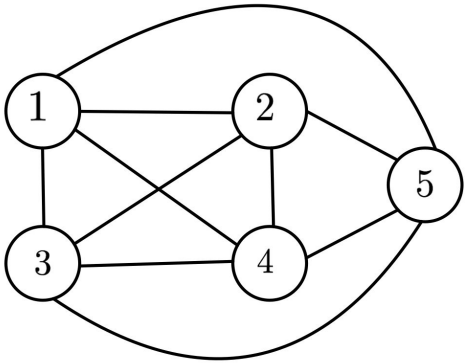
Counting - Enda et eksempel

In an imaginary tournament, 20 participating teams are divided into 4 groups of 5 teams. In each group, all 5 teams play once against each other, and the best team of each group progresses to the next round.

a) How many games are played in total during the first round of the tournament? Explain your answer.

b) For a given division of teams into groups, how many combinations of 4 teams (group winners) exist that can appear in the second tournament stage? Your answer can contain factorial or power expressions. Explain your answer.

- Vi har 5 lag per fordeling, alle må spille mot hverandre.
    - Vi kan tegne en graf der nodene representerer lag, og kantene representerer spill.



- 10 kanter = 10 spill
- Vi har også 4 grupper med 5 lag hver
    - Sum Regelen sier da at vi har 10 + 10 + 10 + 10 = 40 spill i første runde.

Alternativt:

$$\binom{5}{2} + \binom{5}{2} + \binom{5}{2} + \binom{5}{2} = 4\binom{5}{2} = 4 \cdot 10 = 40$$

b) Hvert lag kan vinne i en gitt lagfordeling: 5 muligheter.

For hver mulig vinner I fordeling 1, har vi 5 mulige vinnere i fordeling 2.

Med produkt regelen kan vi konkludere at det er:

$5 \cdot 5 \cdot 5 \cdot 5 = 5^4$ kombinasjoner av vinnere

Sannsynlighet

Sannsynlighet er bare applied counting

- Utfallsrom
  - S : alle mulige utfall
  - E : alle gunstige utfall
  - $E \subseteq S$

$$P(E) = \frac{|E|}{|S|}$$

- $P(E) \geq 1$: alltid.
- $P(S) = 1$
- $P(\bar{E}) = 1 - P(E)$

To viktige regler:

$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$

$$P(A \cap B) = P(A)P(B)$$

Sannsynlighet - Prob. Theory

- Husk P(S) = 1
  - vi definerer sannsynligheten for hver $s \in S$ med:
    - $0 \leq P(s) \leq 1$
    - summen av disse sannsynlighetene blir $\sum_{s \in S} P(s) = 1 \rightarrow P(S) = 1$
- Husk $P(E) = \frac{|E|}{|S|}$
- Eksempel: A pair of dice is loaded. The probability that a 4 appears on the first die is $2/7$, and the probability that a 3 appears on the second die is $2/7$. Other outcomes for each die appear with probability $1/7$. What is the probability of 7 appearing as the sum of the numbers when the two dice are rolled?

Sannsynlighet - Prob. Theory - Oppgave

Vi vet at P(4) = 2/7 ved første kast og P(3) = 2/7 ved andre kast.

- Finn alle sannsynligheter hvor vi får 7:
  - P(6)P(1) = 1/7 * 1/7
  - P(5)P(2) = 1/7 * 1/7
  - P(4)P(3) = 2/7 * 2/7
  - P(1)P(6) = 1/7 * 1/7
  - P(2)P(5) = 1/7 * 1/7
  - P(3)P(4) = 1/7 * 1/7
- Summer opp for å få 4/49.

Relations - Introduksjon

A = {a, b, c} B = {1,2}
A x B = {(a,1),(a,2),(b,1),(b,2),(c,1),(c,2)}

A og B er sett

R er et subsett av A x B

Vi sier at det er en relasjon fra A til B

Hvilke (a,b) elementer vi skal ta med i R er bestemt av relasjonen

Eksempel:

A = alle studenter på informatikk, B = alle emner på UIB

Relasjonen er studenten **a** har tatt emne **b**

(Ole, INF100) er i R om Ole har tatt INF100

(Ole, INF100), (Kari, MNF130), (Ole, MNF130) $\in$ R

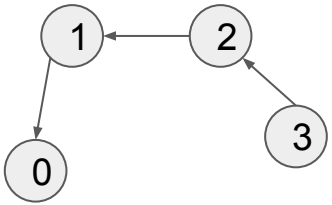
Relations - Introduksjon - Eksempler

Relasjonene under en på settet S = {0, 1, 2, 3}

R_1 = {(a,b) | a = b + 1} = {(1, 0), (2, 1), (3, 2)}

R_2 = {(a,b) | a+b <= 3} = {(0, 0), (0, 1), (0, 2), (0, 3),
(1, 0), (1, 1), (1, 2), (2, 1), (3, 0)}

Vi kan representere relasjoner på forskjellige måter



□← Graf representasjon av R_1

a\b	0	1	2	3
0	1	1	1	1
1	1	1	1	0
2	1	1	0	0
3	1	0	0	0

↑ Matrise representasjon av R_2

Relations - Egenskaper

Vi er ofte ute etter om relasjonen har noen spesielle egenskaper:

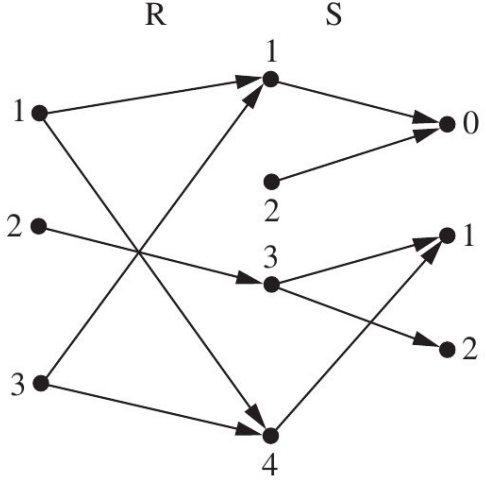
En relasjon kan være:

- **Reflexive**: if $(a, a) \in R$ for every $a \in A$
- **Symmetric**: if $(b, a) \in R$ whenever $(a,b) \in R$, for all $a, b \in A$
  - **Antisymmetric**: for all $a, b \in A$, if $(a,b) \in R$ and $(b,a) \in R$, then $a = b$
- **Transitive**: for all $a, b, c \in R$, if $(a,b) \in R$ and $(b, c) \in R$, then $(a,c) \in R$

Relations - Composite relation

Let R be a relation from a set A to a set B and S a relation from B to a set C.

The composite of R and S is the relation consisting of ordered pairs (a,c) where $a \in A$, $c \in C$, and for which there exists an element $b \in B$ such that $(a, b) \in R$ and $(b, c) \in S$ We denote the composite of R and S by $S \circ R$



$1 \to 1 \to 0 \quad (1, 0)$
$1 \to 4 \to 1 \quad (1, 1)$
$2 \to 3 \to 1 \quad (2, 1)$
$2 \to 3 \to 2 \quad (2, 2)$
$3 \to 1 \to 0 \quad (3, 0)$
$3 \to 4 \to 1 \quad (3, 1)$

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

S = {(1,0),(2,0),(3,1),(3,2),(4,1)}

S ◦ R = {(1,0),(1,1),(2,1),(2,2),(3,0),(3,1)}

Relations - Eksempel 1 og 2: Ekvivalensrelasjoner

Eksempel 1:

$x,y \in \mathbb{R}$: $(x, y) \in R$ iff. $x = 1 \vee y = 1$

$(1, r), (r, 1) \in R, r \in \mathbb{R}$

Eksempel 2:

R: {(1,2),(1,3),(2,3),(2,4),(3,1)}

S: {(2,1),(3,1),(3,2),(4,2)}

R ◦ S: {(2,2),(2,3),(3,2),(3,3),(3,4),(4,3),(4,4)

Eksempel 1:

r = 1, $(1,1) \in R$

r = 5, $(1,5) \in R$, $(5, 1) \in R$

Her ser vi at R er symmetrisk, men ikke reflexiv siden (5, 5), (r,r) ikke finnes i R $\forall r \in \mathbb{R}$. Den er heller ikke transitiv, siden (r,1),(1,r) -> (r,r)

Eksempel 2:

R ◦ S er en relasjon på settet: {1,2,3,4}

Reflexive: ❌

Symmetric: ✓

Anti-symmetric: ❌

Transitive: ✓

Relations - Eksempel 3 og 4

Eksempel 3:

$a,b \in \mathbb{Z}$: $(a, b) \in R \leftrightarrow b-a = 1$

Hva er $R \circ R$?

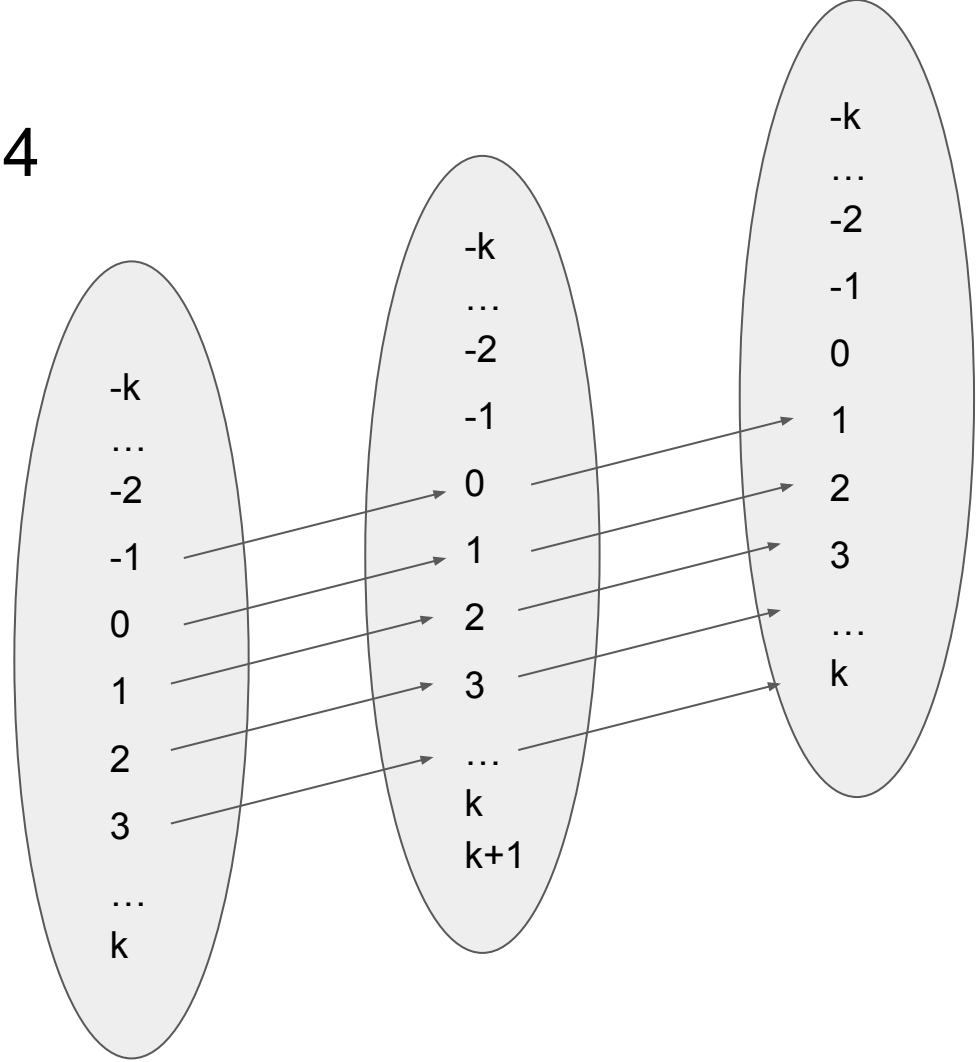
$(a, c) \in R \circ R \leftrightarrow \exists\ b\ [\ (a,b) \in R \land (b,c) \in R\ ]$

$\equiv (a, c) \in R \circ R \leftrightarrow \exists\ b\ [\ (b - a = 1) \land (c - b = 1)\ ]$

$b = c - 1$, setter inn for b i ↑

$(a, c) \in R \circ R \leftrightarrow (c - 1 - a = 1)$

$(a, c) \in R \circ R \leftrightarrow (c - a = 2)$



Relations - Equivalence relation

Properties of equality:

Reflexive property: a = a

Symmetric property: if a = b, then b = a

Transitive property: if a = b and b = c, then a = c

A relation is an equivalence relation if it is reflexive, symmetric and transitive.

Relations - Equivalence relation - Eksempel

Properties of equality:

Reflexive property: a = a

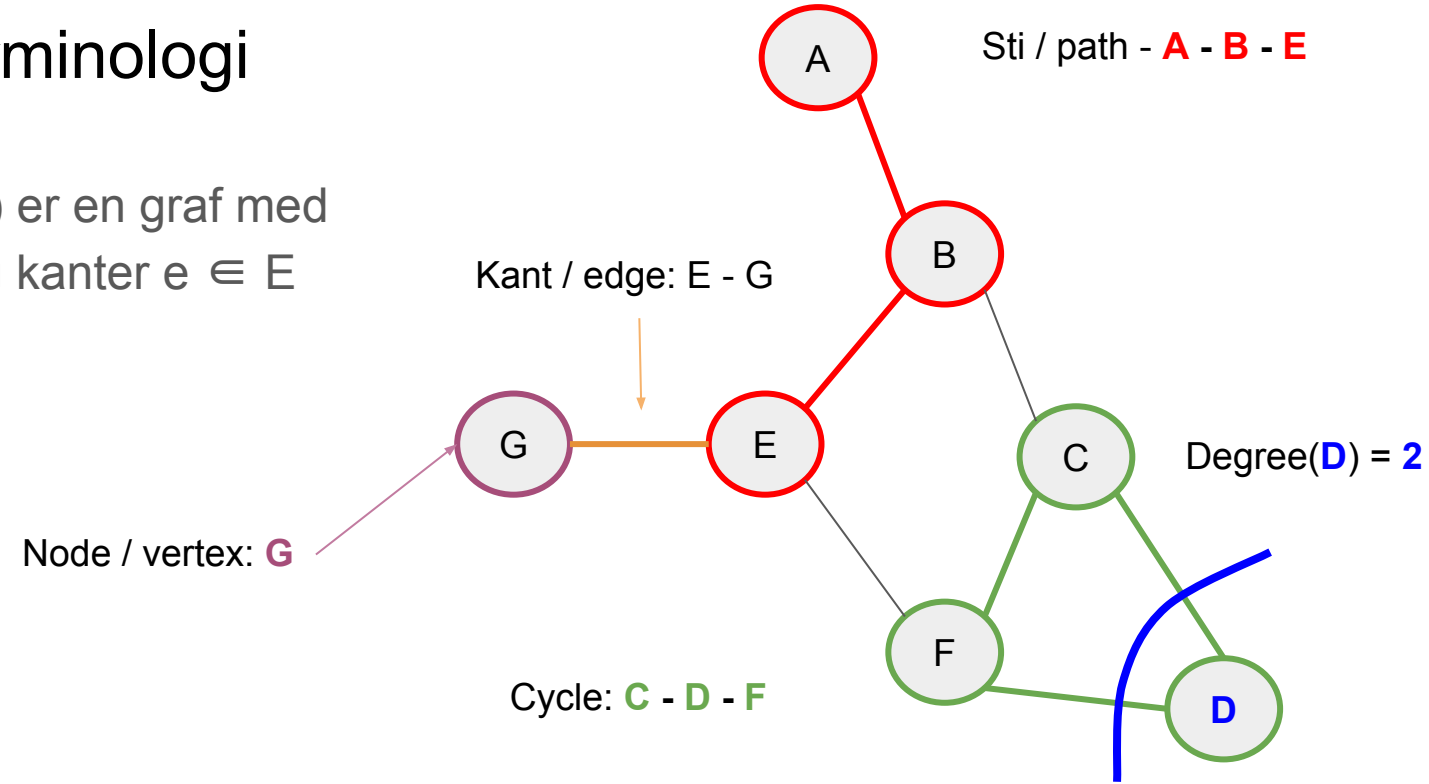
Symmetric property: if a = b, then b = a

Transitive property: if a = b and b = c, then a = c

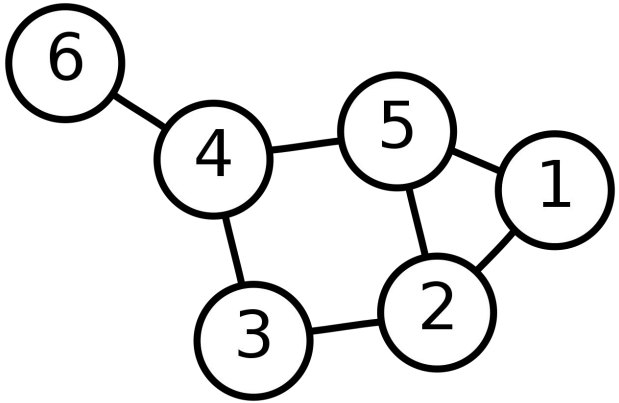
A relation is an equivalence relation if it is reflexive, symmetric and transitive.

Graphs - Terminologi

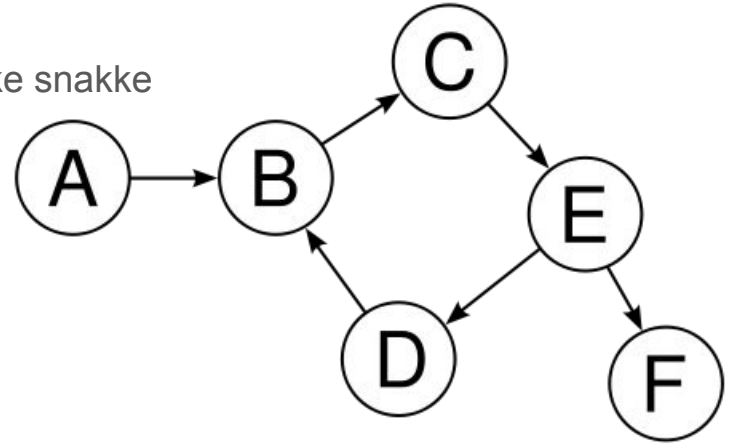
- En Graf G(V, E) er en graf med noder v ∈ V og kanter e ∈ E



Graphs - Terminologi



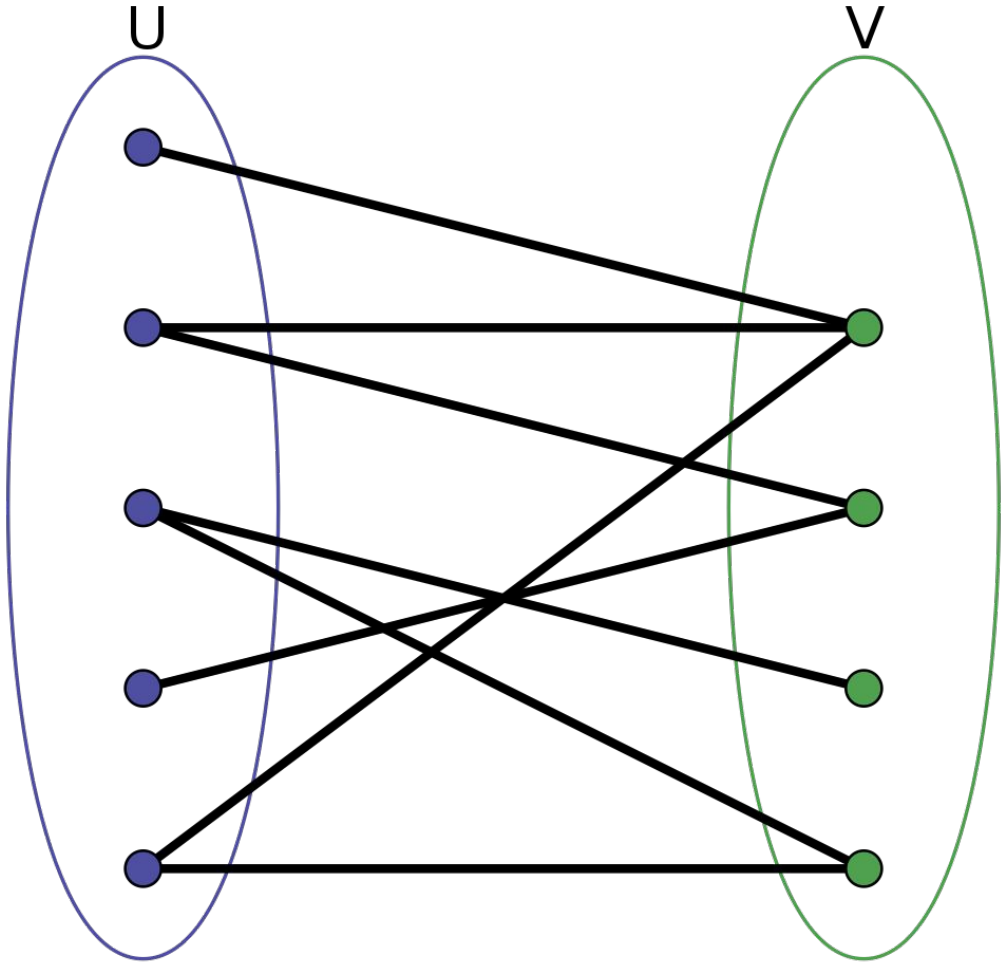
- Undirected
  - Som Grafen på forrige slide.
  - Undirected graphs har ikke en retning på kantene
    - det representerer ofte at to to noder snakker sammen begge veier
- Directed
  - Directed graphs (digraphs) har retning på kantene
  - I denne grafen så kan A snakke til B men b kan ikke snakke til a.



Grafer - Useful terminologi

- Husk Graf G(V, E) består av settet V med noder og settet E med kanter
- Degree deg(v)
  - for en v ∈ V er deg(v) antall kanter ut/inn fra/til v
  - i digraphs så skriver vi deg^-(v) for kanter som går inn mot v (in-degree) og deg^+(v) for kanter som går ut av v (out-degree)
- Neighborhood N(v)
  - nodene som er connected til node v
  - vi snakker sjeldent om neighborhoods i digraphs.
- Bipartite grafer
  - En graf er er bipartit hvir V kan deles opp i to disjunkte sett V_1 og V_2 slik at hver kant kobler sammen kobler sammen en v i V_1 med en v i V_2 slik a, men ikke to noder fra samme sett.
  - Eller By Theorem 4
    - Hvis du kan farge alle nodene i grafen med 2 farger slik at en kant aldri treffer samme farge.

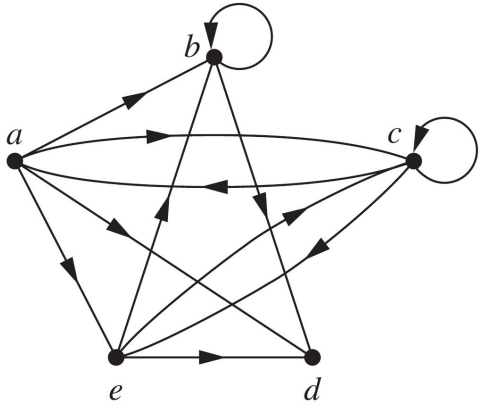
Eksempel - Bipartit graf



Grafer - Representasjoner

- Adjacency list
  - liste av noder og dens naboer
  - for digraphs så lister vi bare nodene gitt node peker til

- Adjacency matrix
  - en nxn matrise med n = |V|
  - a_{i, j} = k hvis node i og j har k kanter mellom seg
  - 0 ellers

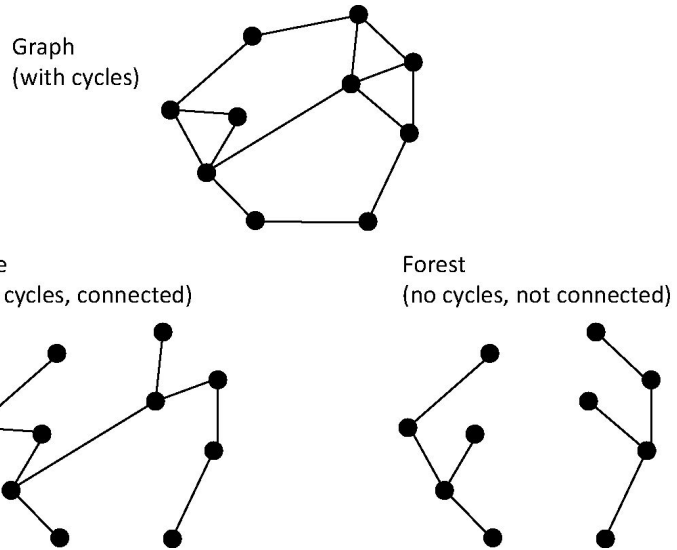


Adjacency matrix for directed graph					
	a	b	c	d	e
a	0	1	1	1	1
b	0	1	0	1	0
c	1	0	1	0	1
d	0	0	0	0	0
e	0	1	1	1	0

Adjacency list for directed graph	
Initial vertex	Terminal vertices
a	b, c, d, e
b	b, d
c	a, c, e
d	
e	b, c, d

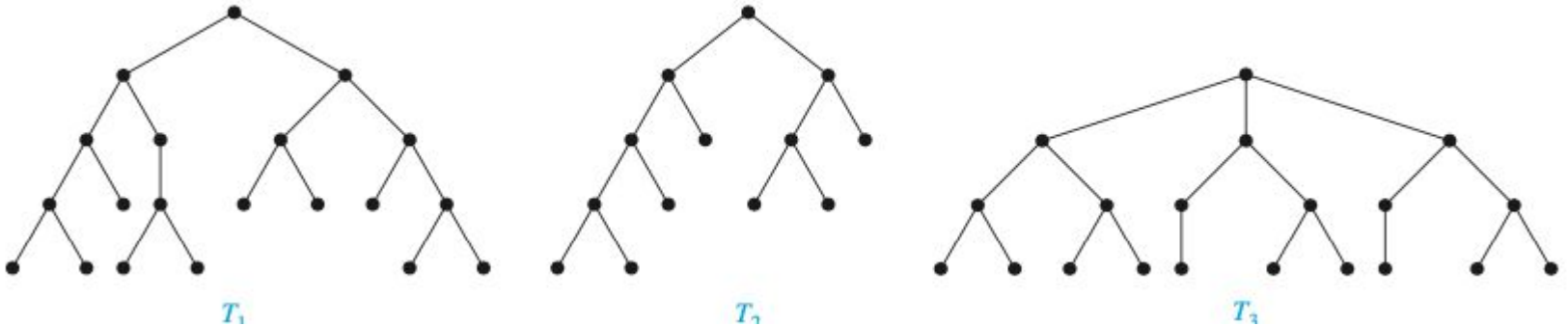
Trær

- Definisjon Tre:
  - An undirected graph in which any two vertices are connected by exactly one path.
  - Eller: En sammenknyttet asyklisk graf.
- Definisjon Forest:
  - En graf som er en union av disjunkte trær.
  - En disconnected asyklisk graf
- Rootet tree:
  - Et tre med en rot node.
  - Binære trær (2-ary)
  - Ternære (3-ary)
  - …
  - n-ary trær



Trær - Egenskaper

- Balansert tre:
  - Alle leaves er på samme høyde.



Trær - Egenskaper

- Et tre med n noder har n-1 kanter

Theorem:

A full m-ary tree with

(i) n vertices has $i = (n-1)/m$ internal vertices and $l = [(m-1)n+1]/m$ leaves,
(ii) i internal vertices has $n = mi+1$ vertices and $l = (m-1)i+1$ leaves,
(iii) l leaves has $n = (ml-1)/(m-1)$ vertices and $i = (l-1)/(m-1)$ internal vertices.

Trær - Eksempel

22. A chain letter starts when a person sends a letter to five others. Each person who receives the letter either sends it to five other people who have never received it or does not send it to anyone. Suppose that 10,000 people send out the letter before the chain ends and that no one receives more than one letter. How many people receive the letter, and how many do not send it out?

Trær - Eksempel

- Siden hver person sender ut 5 brev, har vi et 5-ært tre.
- Node: Representerer person
- Kant: Representerer et brev sendt
- Leaf: Person som ikke sendte et brev
  - Dermed et internal nodes personer som har sendt brev.
- Vi har at 10 000 personer har sendt et brev -> 10 000 internal nodes
  - Fra Theorem: En graf med n kanter har $\frac{n-1}{n} = i$ nterval vertices
    - =: $\frac{n-1}{5} = 10000$
    -
    - = $n - 1 = 50000$
    - = n = 50001 personer i grafen
    - med 10 000 personer som har sendt brev og 50 001 personer som mottok brev
      - => 40 001 sendte ikke brev.

Tree traversal

- Preorder traversal

Let T be an ordered rooted tree with root r. If T consists only of r, then r is the *preorder traversal* of T. Otherwise, suppose that $T_1, T_2, \ldots, T_n$ are the subtrees at r from left to right in T. The *preorder traversal* begins by visiting r. It continues by traversing T_1 in preorder, then T_2 in preorder, and so on, until T_n is traversed in preorder.

- Inorder traversal

Let T be an ordered rooted tree with root r. If T consists only of r, then r is the *postorder traversal* of T. Otherwise, suppose that $T_1, T_2, \ldots, T_n$ are the subtrees at r from left to right. The *postorder traversal* begins by traversing T_1 in postorder, then T_2 in postorder, …, then T_n in postorder, and ends by visiting r.

.

Tree Traversal

- Postorder Traversal

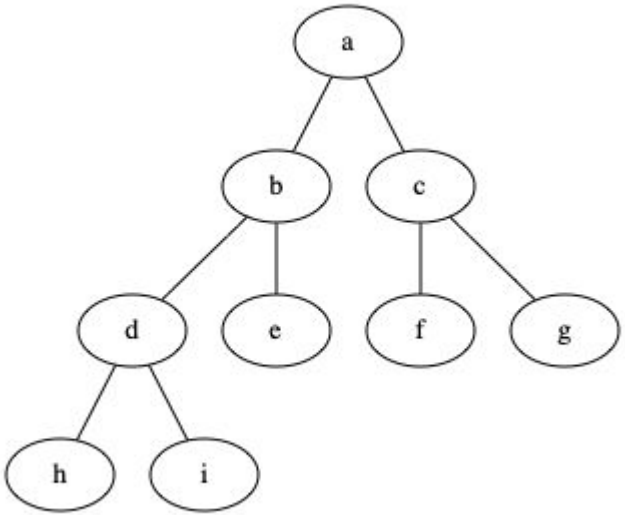
Let T be an ordered rooted tree with root r. If T consists only of r, then r is the *postorder traversal* of T. Otherwise, suppose that $T_1, T_2, \ldots, T_n$ are the subtrees at r from left to right. The *postorder traversal* begins by traversing T_1 in postorder, then T_2 in postorder, $\ldots$, then T_n in postorder, and ends by visiting r.

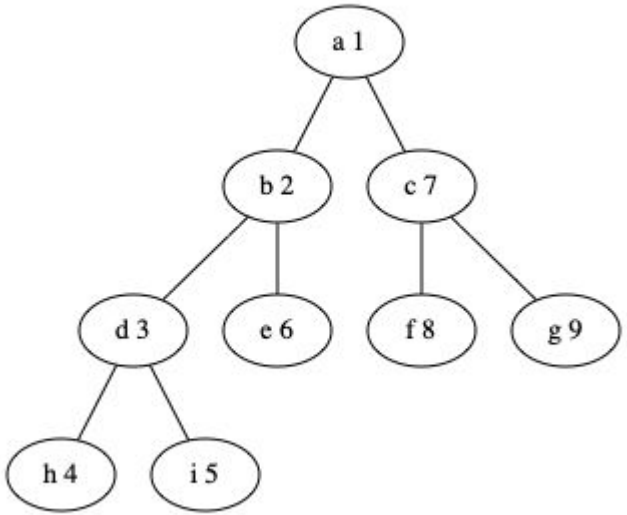
Preorder traversal - Eksempel

ALGORITHM 1 Preorder Traversal.

procedure *preorder*(T: ordered rooted tree)
r := root of T
list r
for each child c of r from left to right
    $T(c)$:= subtree with c as its root
    preorder($T(c)$)

Preorder traversal

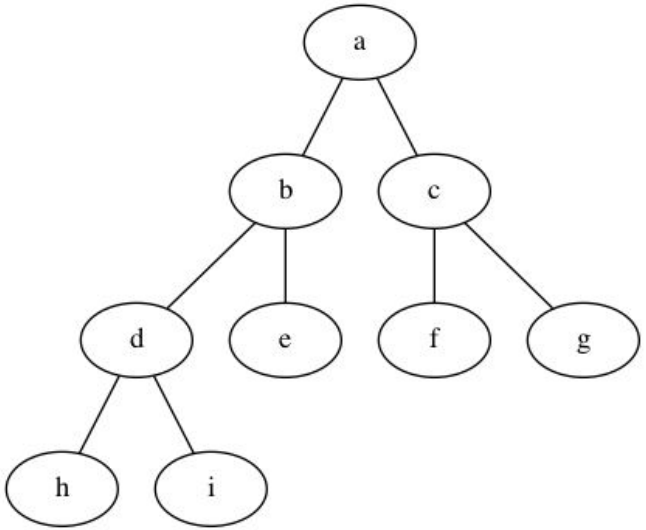


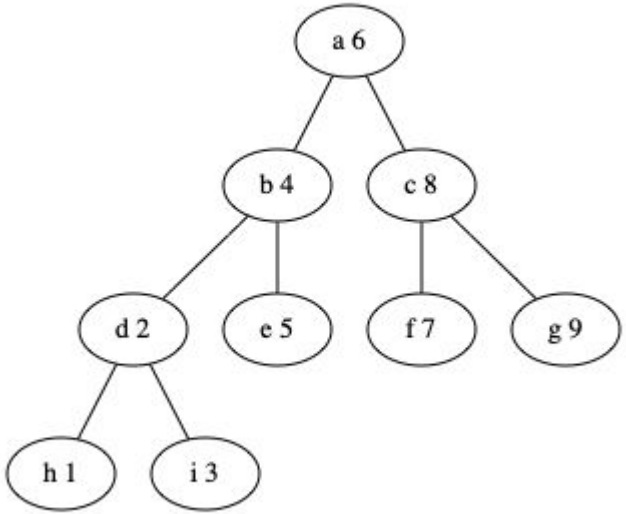


ALGORITHM 2 Inorder Traversal.

procedure *inorder*(T: ordered rooted tree)
r := root of T
if r is a leaf **then** list r
else
  l := first child of r from left to right
  $T(l)$:= subtree with l as its root
  inorder($T(l)$)
  list r
  for each child c of r except for l from left to right
    $T(c)$:= subtree with c as its root
    inorder($T(c)$)

Inorder traversal

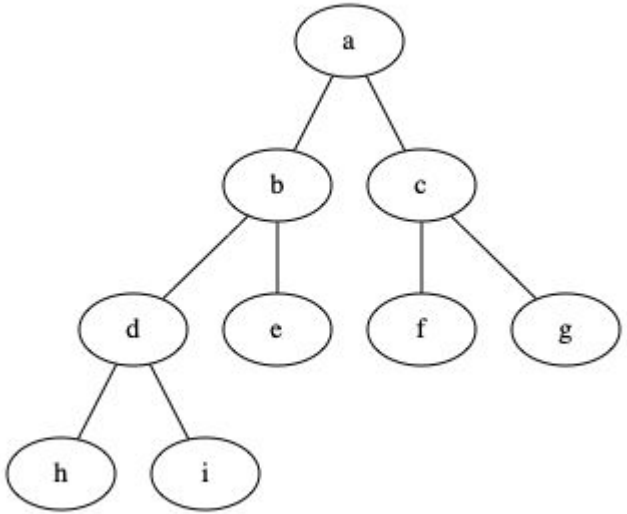


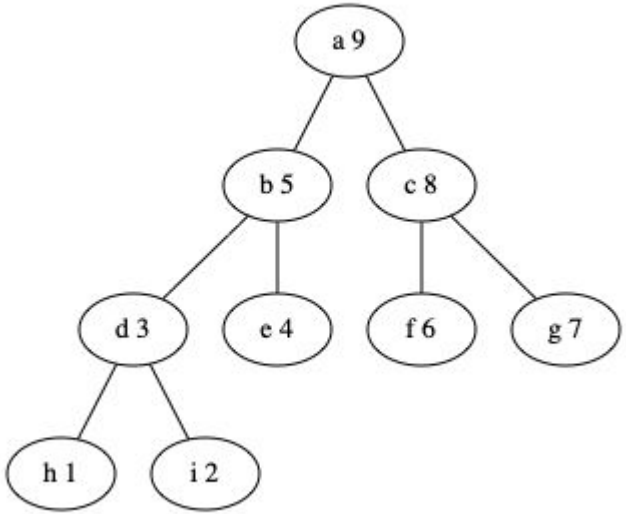


ALGORITHM 3 Postorder Traversal.

procedure *postorder*(T: ordered rooted tree)
r := root of T
for each child c of r from left to right
    $T(c)$:= subtree with c as its root
    postorder($T(c)$)
list r

Postorder traversal -

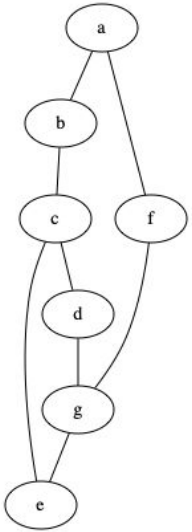


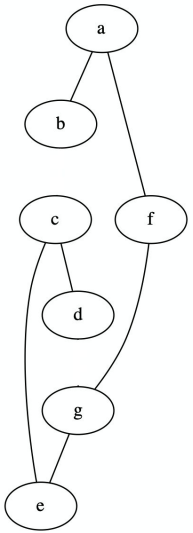


Spanning trees.

- Husk at et tre er en asyklisk og sammenkoblet graf.
- Et Spanning Tree $T \subseteq G$ er et tre som “utspenner” G (besøker alle nodene uten å lage sykluser)

- Algoritmer for å finne Spanning trees:
  - Depth First Search (DFS)
  - Breadth First Search (BFS)





DFS

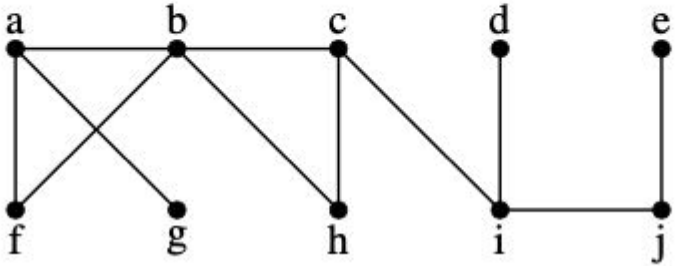
ALGORITHM 1 Depth-First Search.

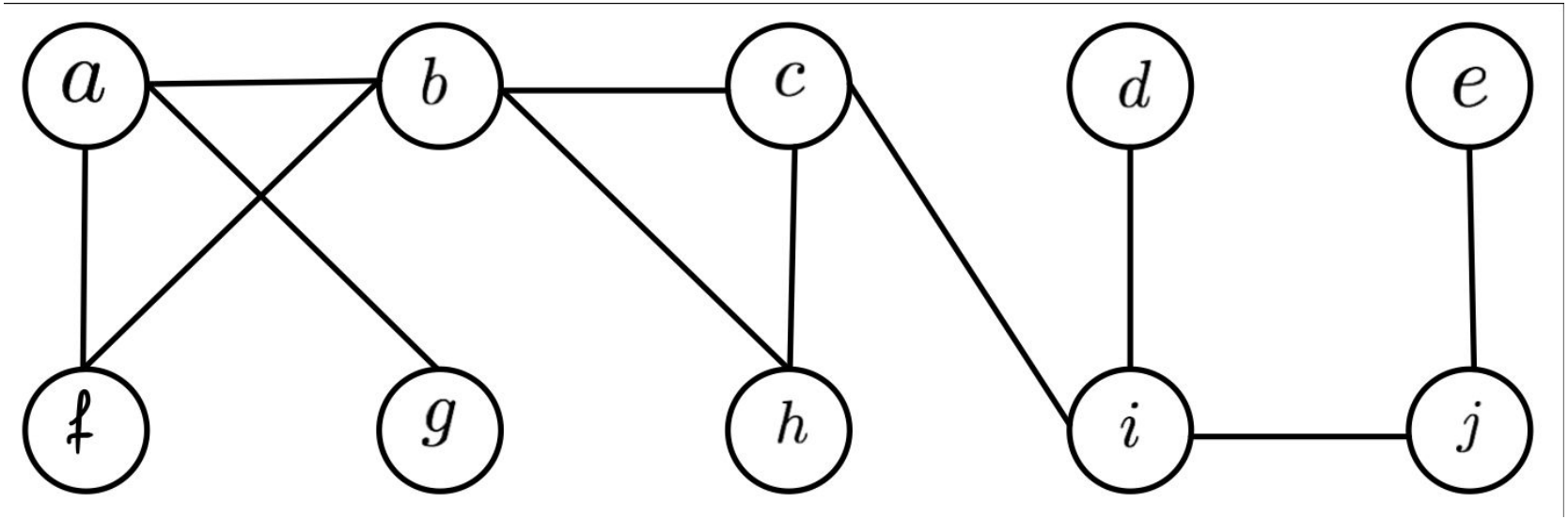
procedure *DFS*(G: connected graph with vertices $v_1, v_2, \ldots, v_n$)
$T :=$ tree consisting only of the vertex v_1
visit(v_1)

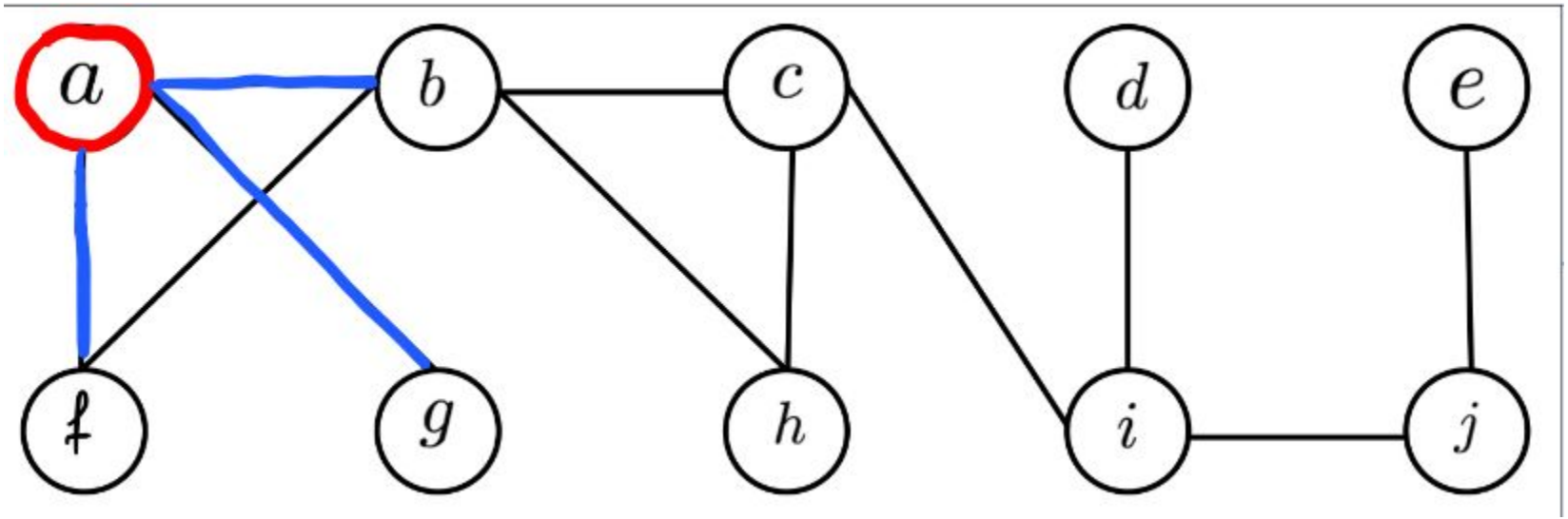
procedure *visit*(v: vertex of G)
for each vertex w adjacent to v and not yet in T
    add vertex w and edge $\{v, w\}$ to T
    visit(w)

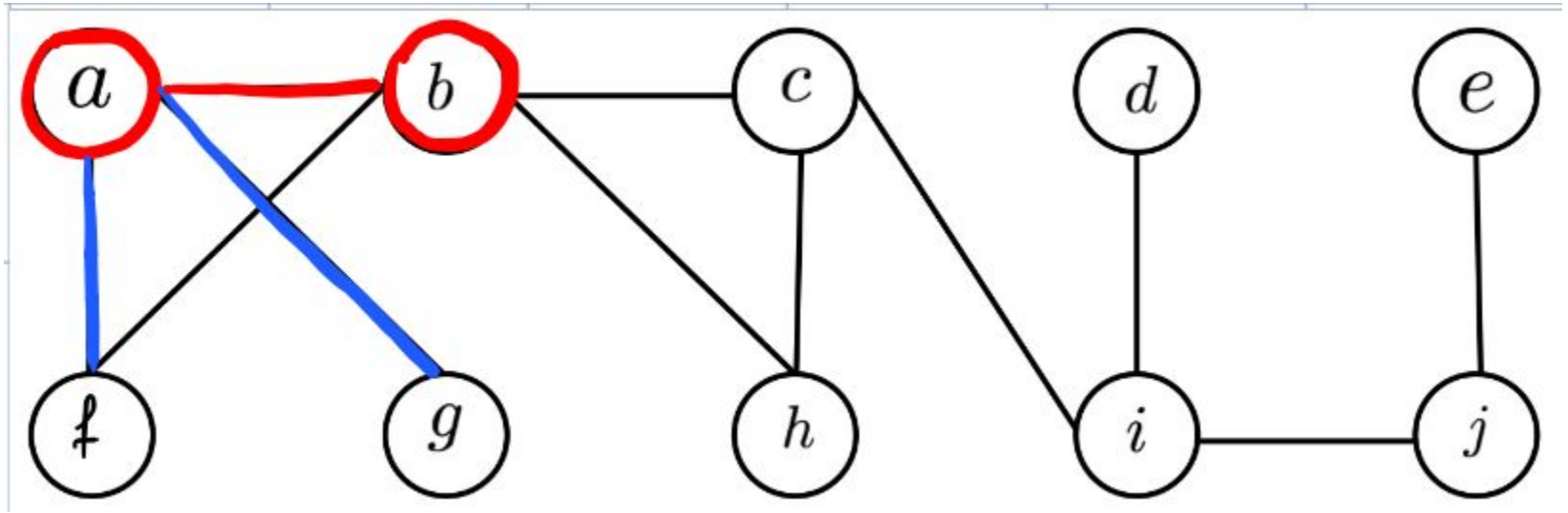
DFS Eksempel.

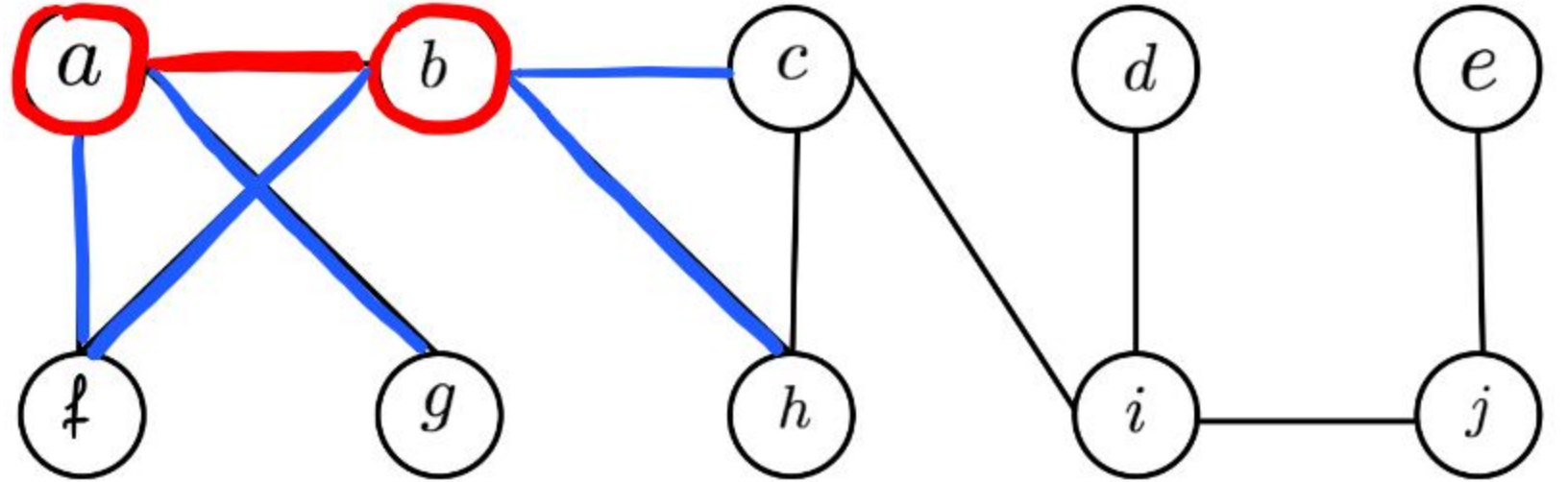
For the graph shown, find a spanning tree using a depth-first search starting from vertex *a*. Use alphabetical ordering if there are multiple choices for adding the next vertex to the tree.

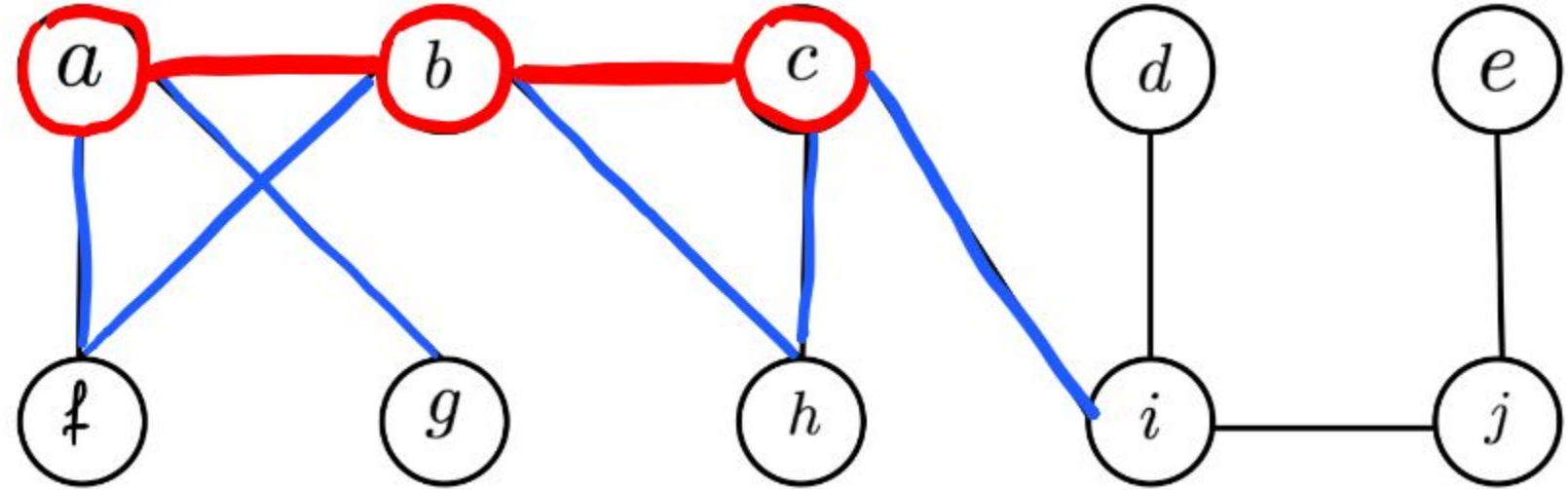


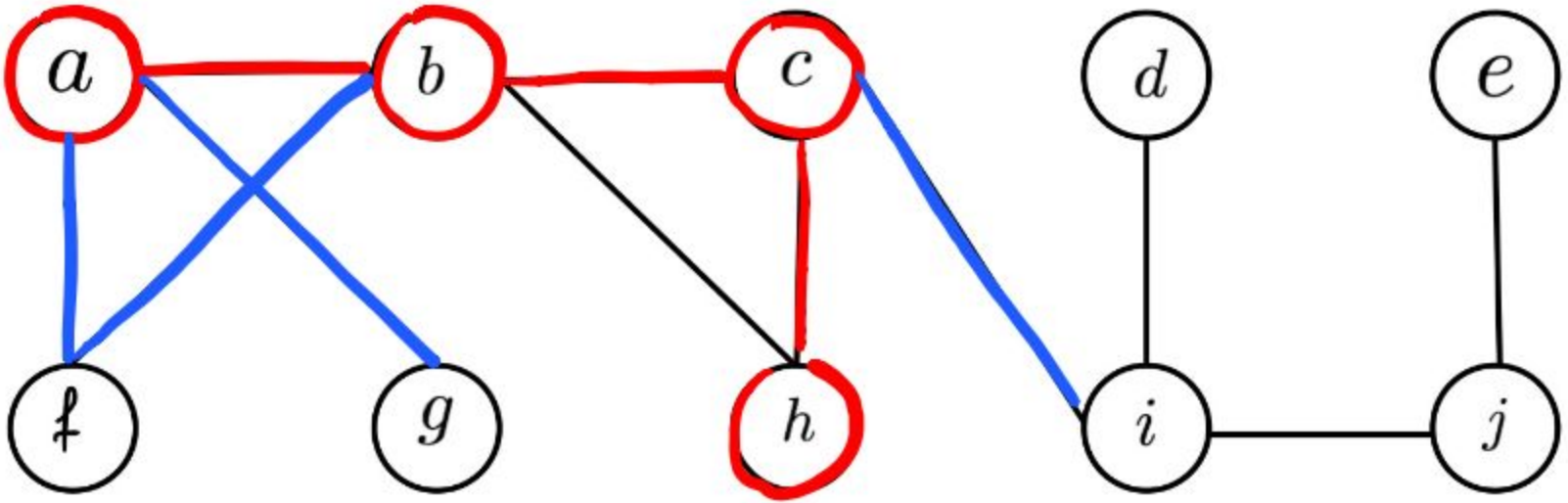

a
b
c
d
e
f
g
h
i
j

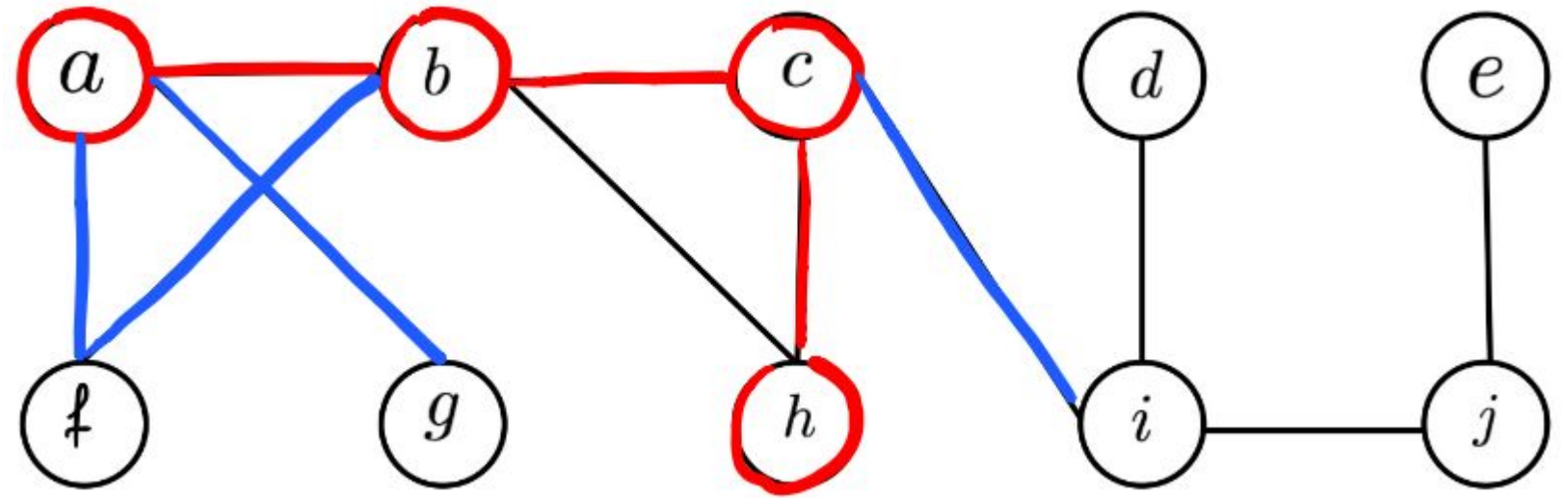

a
b
c
d
e
f
g
h
i
j

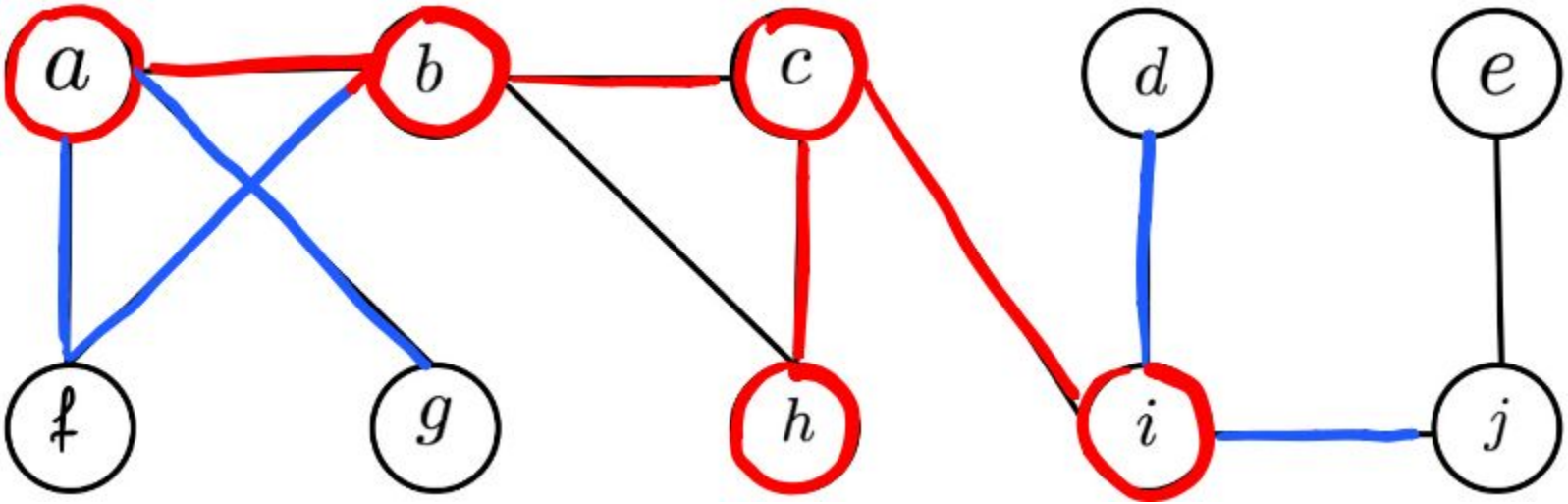

a
b
c
d
e
f
g
h
i
j

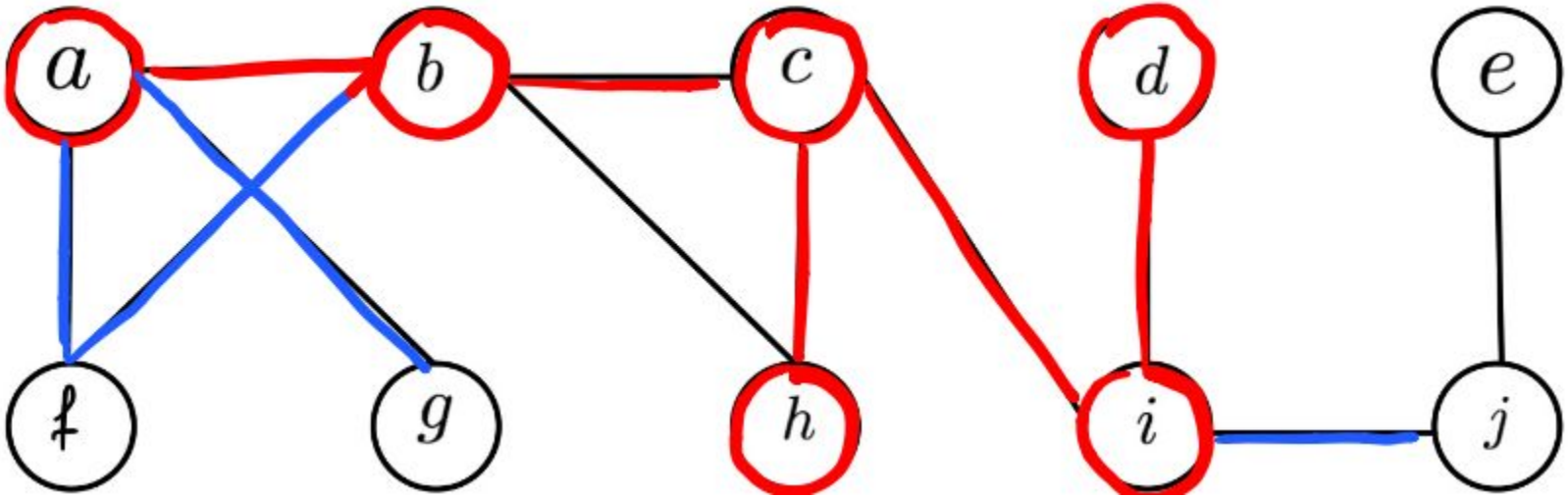

a
b
c
d
e
f
g
h
i
j

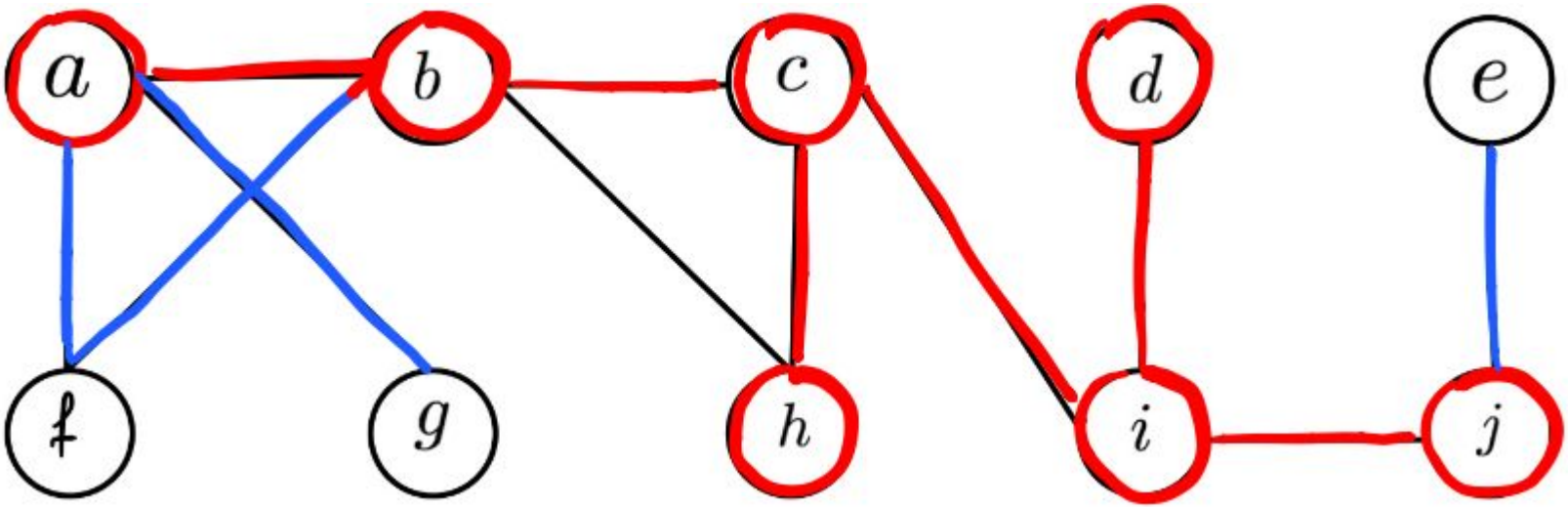

a
b
c
d
e
f
g
h
i
j

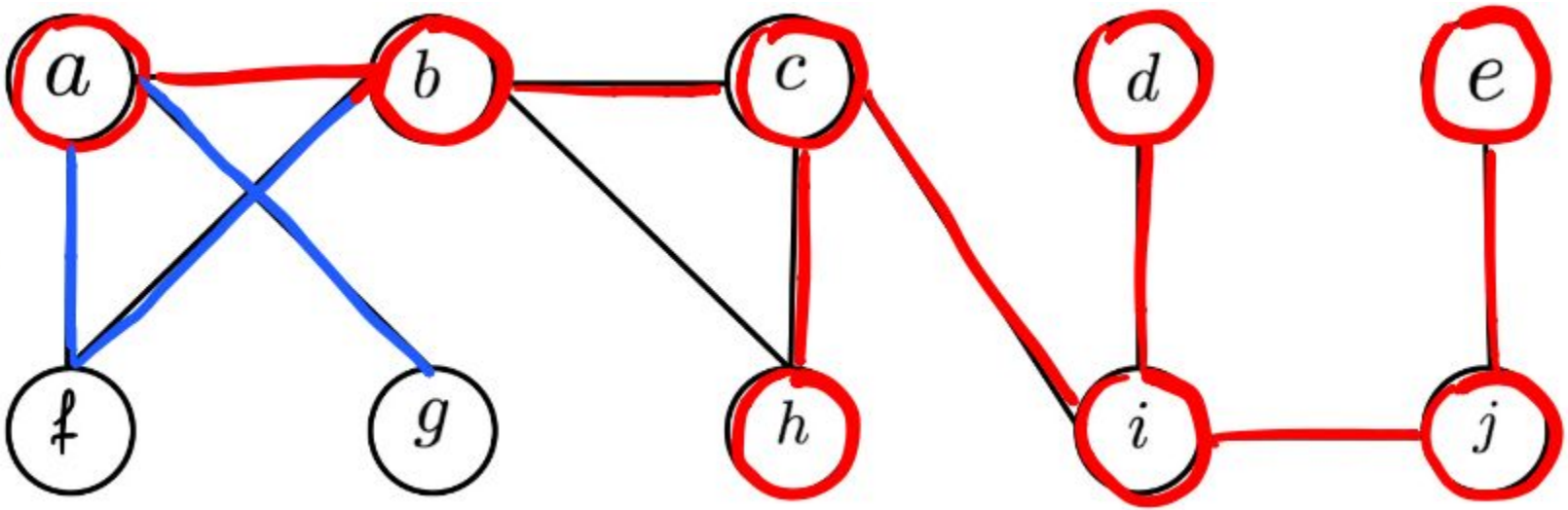

a
b
c
d
e
f
g
h
i
j

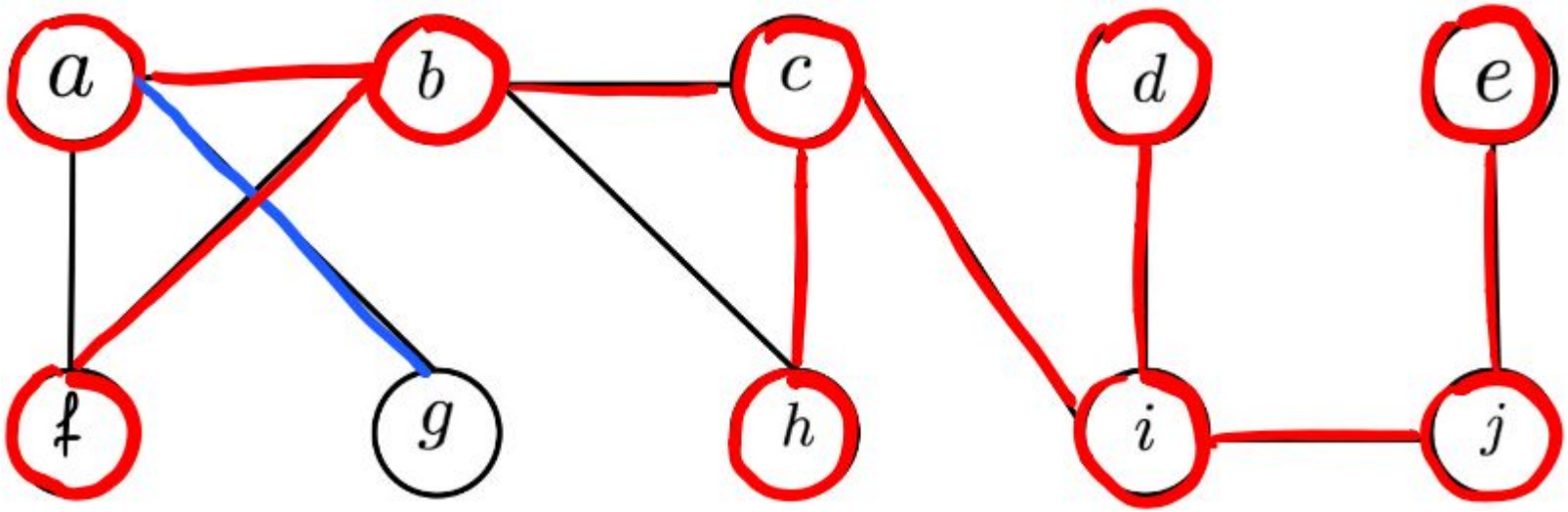

a
b
c
d
e
f
g
h
i
j

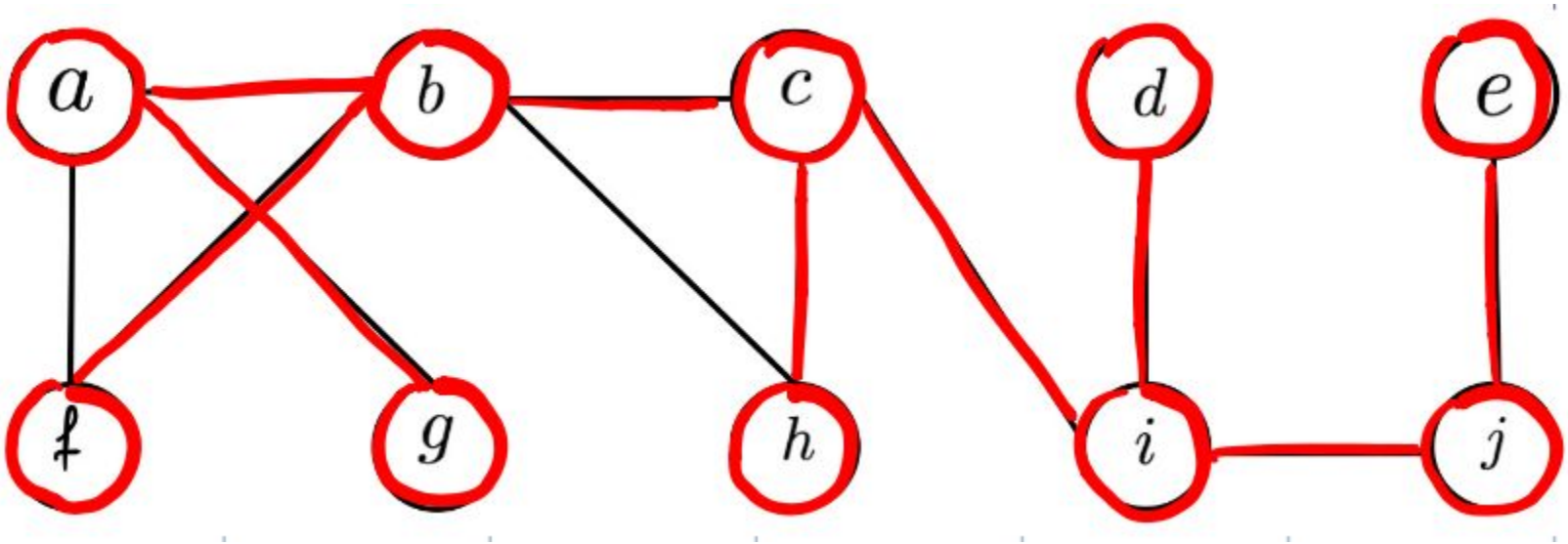

a
b
c
d
e
f
g
h
i
j


a
b
c
d
e
f
g
h
i
j


a
b
c
d
e
f
g
h
i
j


a
b
c
d
e
f
g
h
i
j


a
b
c
d
e
f
g
h
i
j

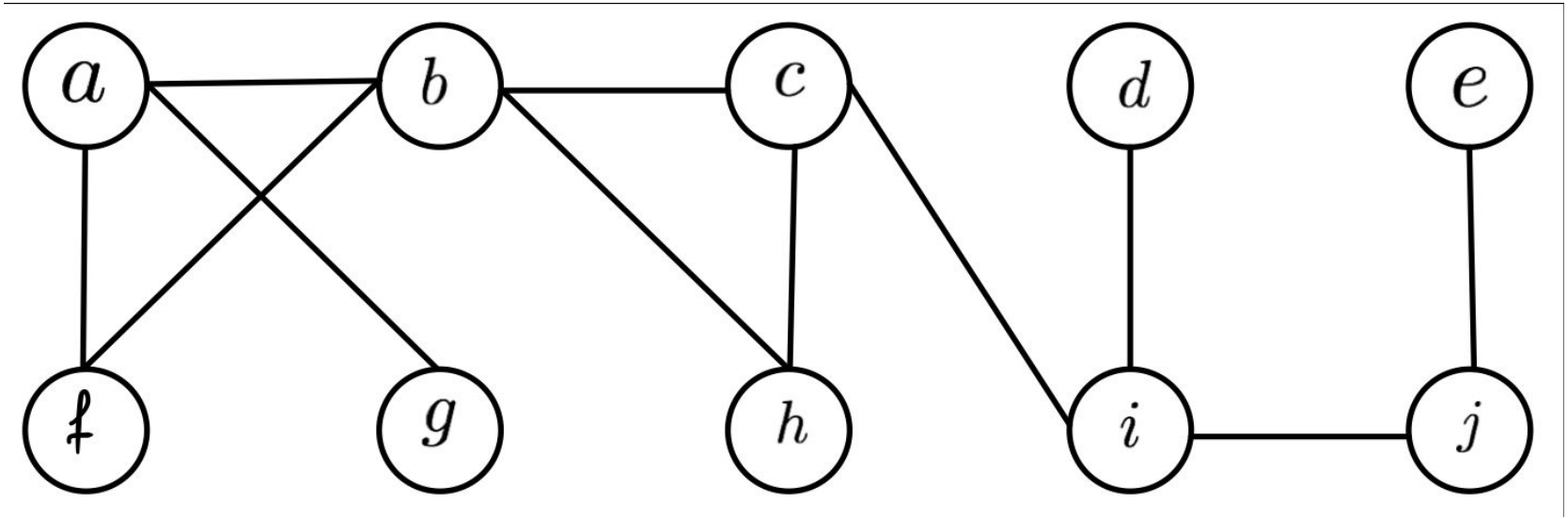

a
b
c
d
e
f
g
h
i
j

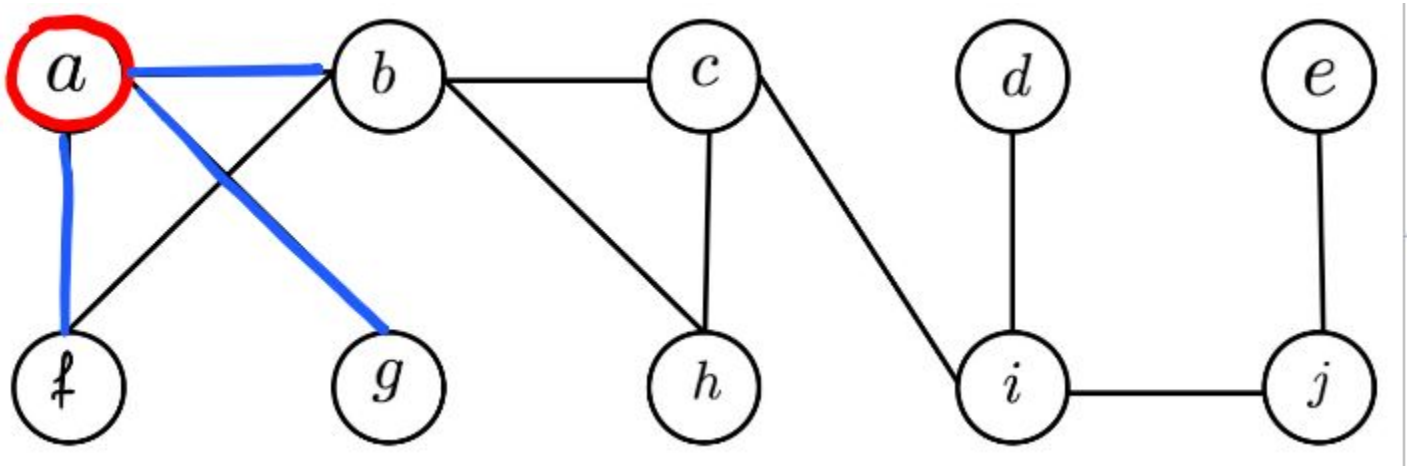
BFS

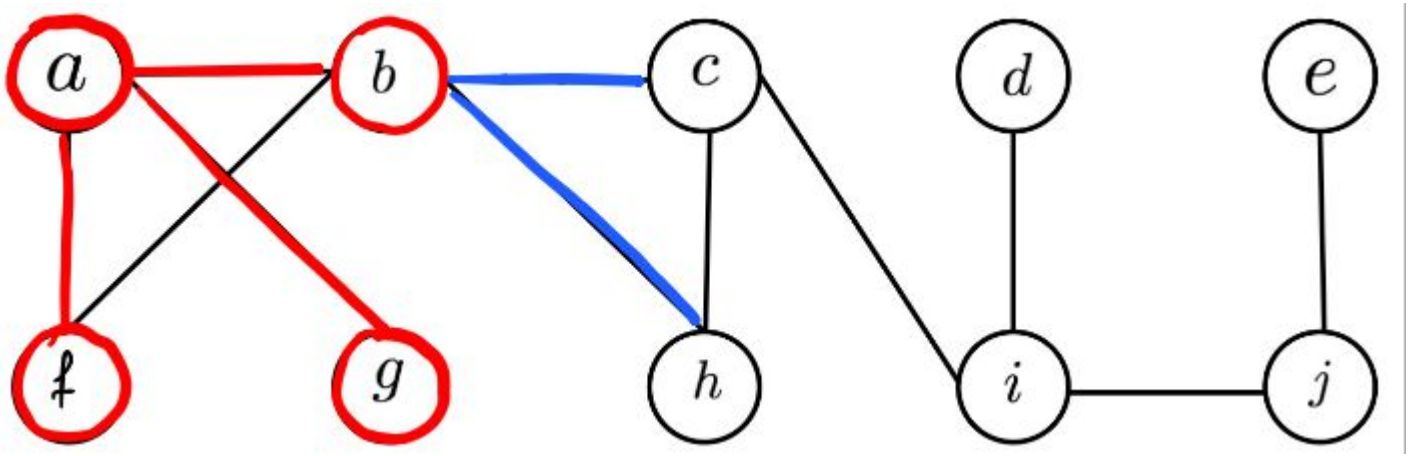
ALGORITHM 2 Breadth-First Search.

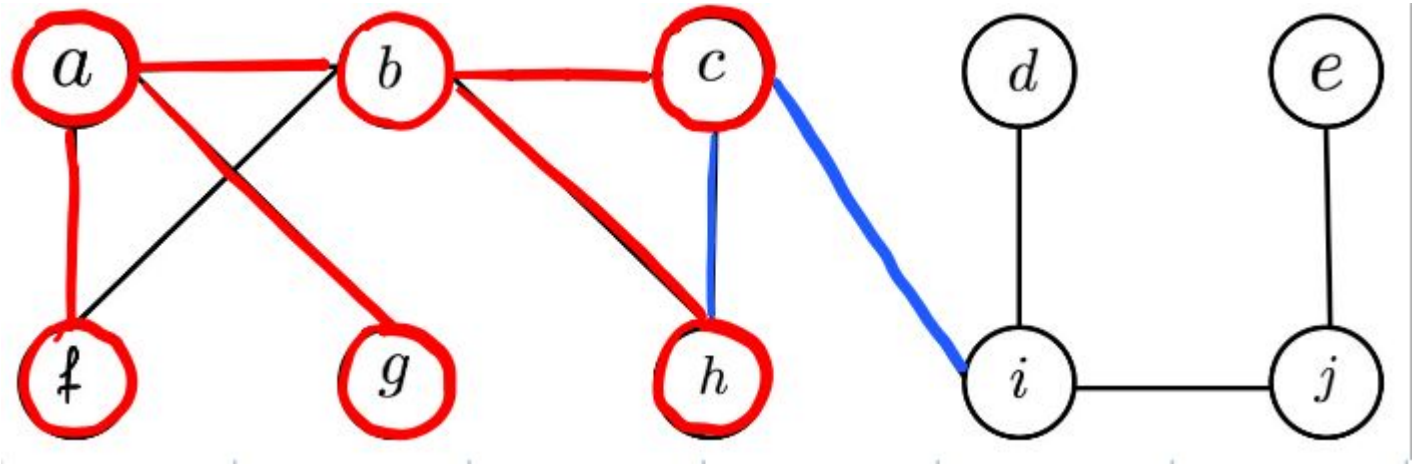
procedure *BFS*(*G*: connected graph with vertices $v_1, v_2, \ldots, v_n$)
T := tree consisting only of vertex v_1
L := empty list
put v_1 in the list L of unprocessed vertices
while L is not empty
  remove the first vertex, v, from L
  for each neighbor w of v
    if w is not in L and not in T **then**
      add w to the end of the list L
      add w and edge $\{v, w\}$ to T

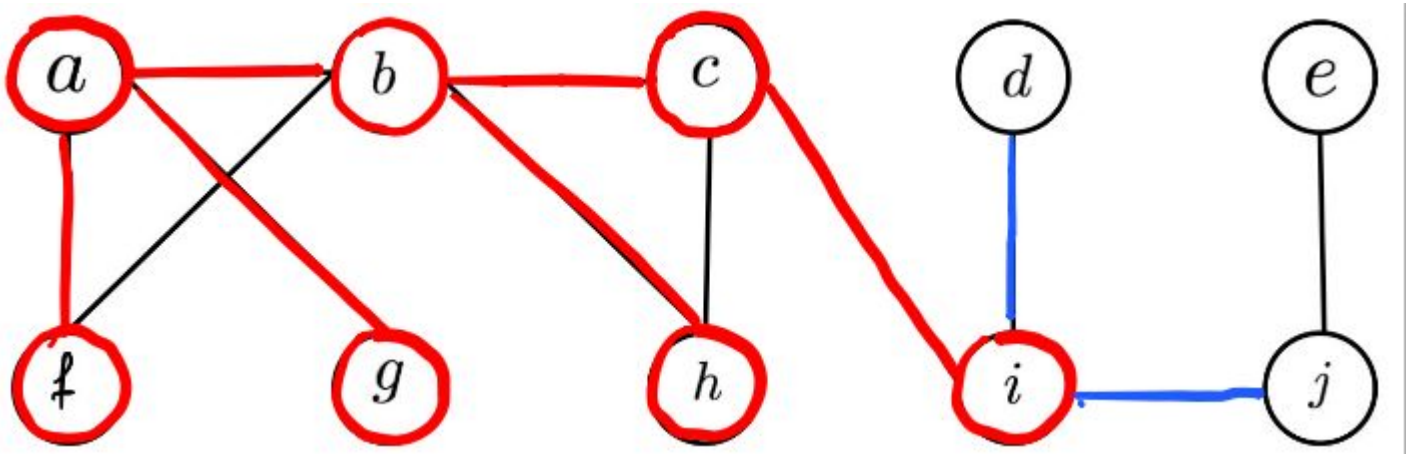
BFS

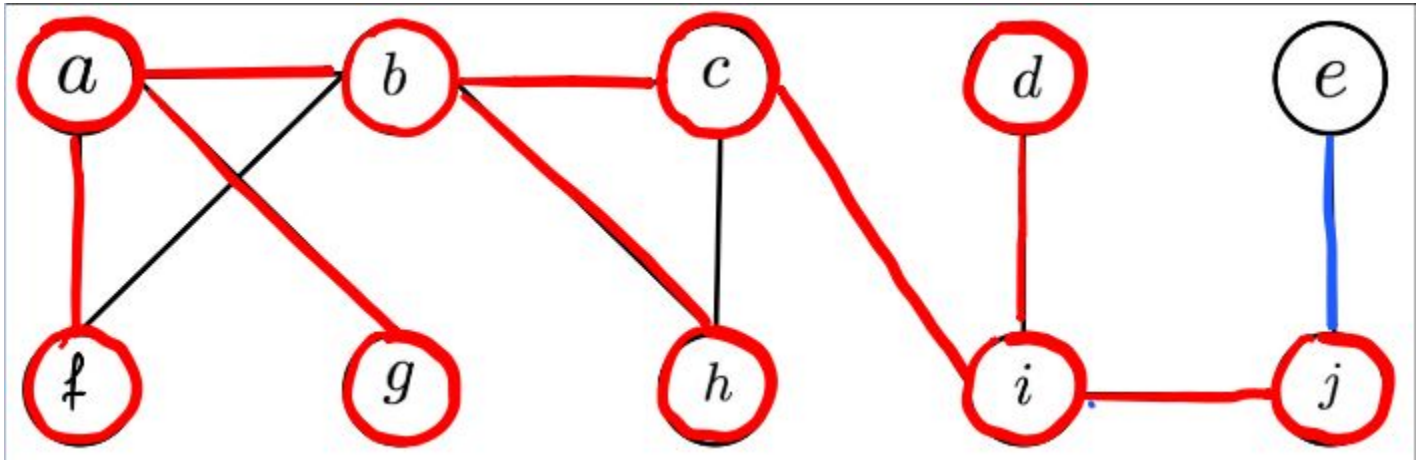


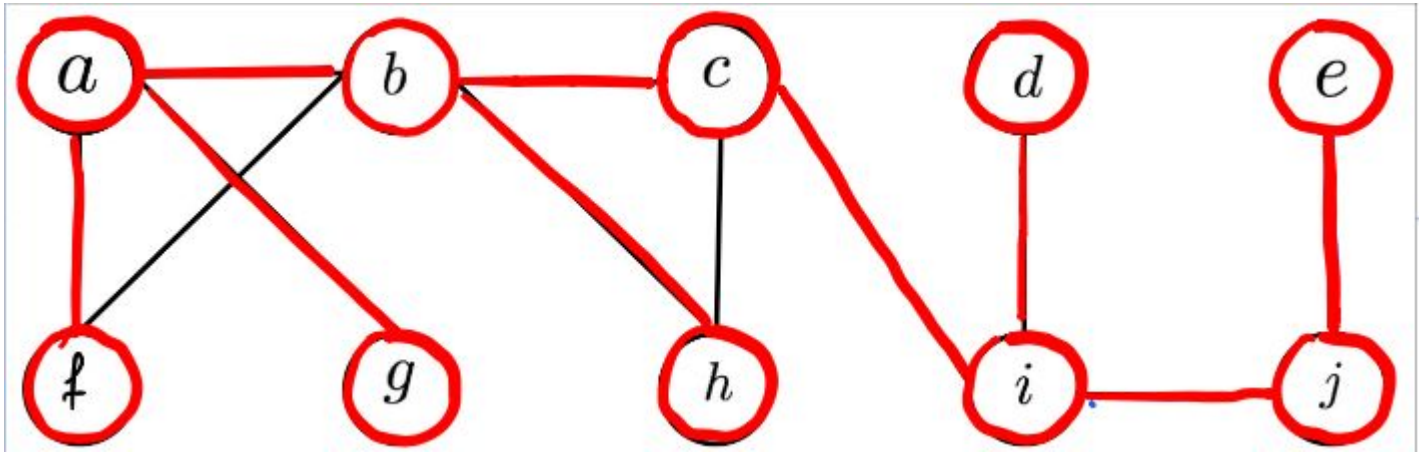

a
b
c
d
e
f
g
h
i
j


a
b
c
d
e
f
g
h
i
j


a
b
c
d
e
f
g
h
i
j


a
b
c
d
e
f
g
h
i
j


a
b
c
d
e
f
g
h
i
j


a
b
c
d
e
f
g
h
i
j