Definition.
Et træ er en sammenhængende ikke-orienteret graf uden simple kredse.

En skov er en ikke-orienteret graf uden simple kredse.

Et træ er altå en sammenhængende skov.

En skov er en graf hvor hver sammenhængskomponent er en skov.

Egenskaber.
Hvis T er en ikke-orienteret graf med n punkter så er følgende udsagn ækvivalente:

- T er et træ.
- For ethvert par af punkter u og v i T er der en entydig simpel vej fra u til v i G.
- T er sammenhængende og har $n-1$ kanter.
- T er uden simple kredse og har $n-1$ kanter.

Træer med rod r.

v: et punkt i træet.
Hvis $v \neq r$ så er der en entydig nabo til v hvis afstand fra r er 1 mindre end v's afstand fra r. Dette punkt kaldes v's forælder.

De øvrige naboer til v (Hvis $v = r$: alle naboer til v.) har afstand fra r: 1 større end v's afstand fra r. Disse punkter kaldes v's børn.

Punkter uden børn kaldes blade. Andre punkter er indre.

m: positivt helt tal.
Et m-ært træ er et træ med rod hvor hvert punkt har højst m børn.
Hvis hvert indre punkt har præcis m børn så er det et fuldt m-ært træ.

Et 2-ært træ kaldes binært.

Sætning 3+4. Et fuldt m-ært træ med i indre punkter, ℓ blade og i alt $n = i + \ell$ punkter opfylder:

$$n = mi + 1 = \frac{m\ell - 1}{m-1} \text{ og } i = \frac{\ell - 1}{m-1}$$

Definition.
Lad G være en ikke-orienteret graf.
Hvis T er et træ, som er delgraf af G og indeholder alle G's punkter så siger vi at T er et udspændende træ i G.

Sætning.
En ikke-orienteret graf G har et udspændende træ
hvis og kun hvis
G er sammenhængende.

Problemer, der løses af algoritme, der svarer JA eller NEJ til inputtet.
Hierarki af sværhedsgrader (m.h.t. kompleksitet):

P:	problemer der løses af hurtig algoritme (polynomiel tid)
NP:	blanding af lette og vanskelige problemer (let at svare JA)
NP-komplet:	meget vanskelige problemer

Millon dollar spørgsmål: er $P = NP$?
De fleste gætter: Nej!, men man kan ikke bevise det.

Hvis man finder en polynomiel-tids algoritme, der kan løse bare ét af de tusindvis af kendte NP-komplette problemer, så kan samme algoritme bruges til at løse alle problemer NP i polynomiel tid.

Eksempler på NP-komplette problemer:

Hamilton-kreds
Input: graf G
Spørgsmål: har G en Hamilton-kreds.

TSP
Input: vægtet komplet graf K, en konstant C
Spørgsmål: har K en Hamilton-kreds af længde højst C.

k-farvning, $k \geq 3$
Input: graf G
Spørgsmål: har G en farvning med k farver.

Procedure DFS(G: en sammenh. ikke-orienteret graf)
vælg et punkt u
$T :=$ træ bestående af u
visit(u)
{ T er et udspændende træ i G}

Procedure visit(v: et punkt i grafen)
for hver nabo w til v
    if $w \notin T$ **then**
    begin
        tilføj w og $\{v, w\}$ til T
        visit(w)
    end

Procedure Kruskal(G: vægtet graf)
sorter G's kanter $e_1, \ldots, e_m$ efter vægt så $w(e_1) \leq w(e_2) \leq \ldots \leq w(e_m)$.
$T :=$ skov uden kanter
for $i := 1$ **to** m
    if $T \cup \{e_i\}$ ikke har simpel kreds **then**
        $T := T \cup \{e_i\}$
{ T er et minimum vægt udspændende træ.}

Procedure Prim (G: vægtet graf med n punkter)
$T :=$ træ bestående af ét punkt
for $i := 1$ **to** $n - 1$
begin
    $e :=$ kant med minimal vægt mellem et punkt i T og et punkt ikke i T
    Tilføj e og endepunkt til T
end
{ T er et minimum vægt udspændende træ}

Procedure Prim ($G = (V, E)$: vægtet graf med n punkter)

$T :=$ træ bestående af ét punkt v_1

for alle punkter $u \neq v_1$

    if $\{v_1, u\} \in E$ **then** $L(u) := w(v_1, u)$ **else** $L(u) := \infty$

{For $u \notin T$: $L(u) =$ længden af korteste kant fra u til T}

for $i := 1$ **to** $n - 1$

begin

    vælg $u \notin T$ så $L(u)$ er minimal

    $T := T \cup \{u\} \cup \{$korteste $u - T$ kant$\}$

    for alle v hvor $\{u, v\} \in E, v \notin T$

        if $w(u, v) < L(v)$ **then** $L(v) := w(u, v)$

end