En **vægtet graf** er en (ikke-orienteret, simpel) graf $G = (V, E)$ med en vægtfunktion

$$w : E \mapsto \mathbb{R}.$$

Vægten af en kant $e = \{u, v\}$ skrives også $w(e) = w(u, v)$.

Længden af en vej $e_1, e_2, \ldots, e_k$ i en vægtet graf er

$$w(e_1) + w(e_2) + \ldots + w(e_k).$$

Vi kan eventuelt kræve at $w(e) > 0$ for alle kanter $e \in E$ i en vægtet graf.

Procedure Dijkstra($G = (V, E)$: vægtet sh. graf,
a, z: punkter)

{ Det antages at $w(e) > 0$ for alle $e \in E$}

For alle $v \in V$: $L(v) := \infty$

$L(a) := 0$, $S := \emptyset$

while $z \notin S$

    $u :=$ punkt ikke i S, så $L(u)$ er mindst mulig

    $S := S \cup \{u\}$

    For alle v hvor $\{u, v\} \in E$ og $v \notin S$

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

            $L(v) := L(u) + w(u, v)$

return $L(z)$.

{En korteste vej fra a til z har længde $L(z)$}

Invariant:

1. Hvis $v \in S$ så er $L(v)$ længden af en korteste vej fra a til v i G. Denne vej er indeholdt i S.

2. Hvis $v \notin S$ så er $L(v)$ længden af en korteste vej fra a til v, hvor alle vejens punkter er i $S \cup \{v\}$.

Punkterne tilføjes til S i rækkefølge bestemt ved voksende afstand fra a.

Kompleksitet af Dijkstras algoritme:

Lad n være antallet af punkter i input-grafen.

Ved hvert gennemløb af while-løkken tilføjes et punkt til S.
While-løkken gennemløbes altså højst n gange.

Ved hvert gennemløb:
Find i en liste med (højst) n punkter et punkt med mindst L-værdi: $O(n)$

Undersøg hver af u's højst n naboer: $O(n)$

Komplesitet af den samlede algoritme: $O(n^2)$

En **Hamilton-kreds** i en graf er en simpelkreds, der går gennem hver af grafens punkter én gang.

Den handelsrejsendes problem
Travelling Salesman Problem (TSP).

Lad G være en vægtet graf med en Hamilton-kreds.
F.eks. G er komplet graf (altså: en simpel graf med en kant mellem alle par af punkter).

Find en *kortest* Hamilton-kreds i G.

En **Euler-kreds** i en ikke-orienteret graf $G = (V, E)$ er en simpel kreds $e_1, \ldots, e_n$, som bruger alle kanter i G.

Sætning. En sammenhængende multigraf G med mindst to punkter har en Euler-kreds hvis og kun hvis alle punkter i G har lige grad.

Procedure Kreds(G: graf, hvor alle grader er lige,
v: punkt, hvor $\deg(v) \geq 2$)

$K := v$
$u := v$
while $\deg(u) > 0$
    $w :=$ en nabo til u
    Tilføj kanten $\{u, w\}$ til K
    fjern kanten $\{u, w\}$ fra G
    $u := w$
return K

Invariant:

1. K er en vej fra v til u
2. Hvis $u = v$ så har alle punkter lige grad.
3. Hvis $u \neq v$ så er $\deg(u)$ og $\deg(v)$ ulige og alle andre punkter har lige grad.

Procedure Euler (G: sammenhængende graf, hvor alle
punkter har lige grad.)

$v :=$ vilkårligt punkt i G
circuit:=Kreds(G, v)
$H := G$ minus kanterne i circuit
$u := v$
while u har ikke gennemløbet circuit
    if deg(u) > 0 **then**
        subcircuit:=Kreds(H, v)
        $H := H$ minus kanterne i subcircuit
        circuit:=circuit med subcircuit indsat efter u
    $u :=$ næste punkt på circuit
return circuit

{circuit er en Euler-kreds}

Invariant:

- circuit er en kreds i G
- Hver kant i G er enten i H eller i circuit, ikke i begge.
- Alle punkter på circuit før u har grad 0 i H

Antal operationer, der udføres i procedure Kreds, er

$$\text{konstant} \cdot \text{antal kanter i den konstruerede kreds.}$$

Kompleksitet af procedure Euler: O(antal kanter i G).
(Inklusiv den tid, der bruges i procedure Kreds.)

En Euler-vej i en graf er en simpel vej (fra u til v, hvor $u \neq v$) der bruger hver kant én gang.

Sætning 2. Lad G være en sammenhængende graf og lad u og v være punkter i G.
Så har G en Euler-vej fra u til v
hvis og kun hvis
u og v har ulige grad, og alle andre punkter har lige grad.