

Grafalgoritmer

Traversering

Bredde-først søk (FIFO / Queue)

Tar utgangspunkt i en startnode.
Traverserer grafen etter FIFO-prinsippet.
En node fargelegges sort idet alle nabanoder er oppdaget.

Dybde-først søk (LIFO / Stack)

Tar utgangspunkt i en startnode.
Traverserer grafen etter LIFO-prinsippet.
En node fargelegges sort idet alle etterkommere er traversert.

Minimale spenntreer

Prim's algoritme

Tar utgangspunkt i en startnode.
Benytter en prioritetskø som til enhver tid prioriterer letteste kant (grådig) ut fra nåværende spenntre.

Kruskal's algoritme

Sorterer kantene etter stigende vektverdi.
Velger til enhver tid letteste kant (grådig) som ikke danner en sykel mellom ulike deltrær.

Maksimal flyt

Ford Fulkerson

Finner en flytforøkende sti fra kilde til sluk, gitt at dette er mulig.
Oppdaterer flyten tilsvarende flaskehalsen i den flytforøkende stien.
Gjentar stegene ovenfor til det ikke eksisterer en flytforøkende sti.

Edmond's Karp

Implementasjon av **Ford Fulkerson** algoritme hvor BFS benyttes til å finne en flytforøkende sti.

Maksimal bipartitt matching

En *matching* er en delmengde av alle kanter i en graf, slik at hver node er tilknyttet maks én kant fra delmengden. En *maksimal matching* svarer til en matching bestående av et maksimalt antall kanter.
For å finne den maksimale bipartitte matchingen reduserer vi problemet til et flytproblem, for å så finne den maksimale flyten gjennom nettverket.

Korteste vei

Én til alle

DAG-Shortest-Paths

Gjennomfører en topologisk sortering av nodene i grafen.
Itererer gjennom alle noder i grafen i topologisk sortert rekkefølge, og slakker alle utgående kanter (**u**, **v**) fra hver node **u** til dens nabo **v**.
(Forutsetter at grafen ikke inneholder sykler)

Dijkstra's algoritme

Tar utgangspunkt i en startnode.
Benytter en prioritetskø hvor prioritet er angitt ved distansen fra roten.
Ved traversering legges oppdagede noder inn i prioritetskøen.
(Forutsetter ikke-negativ kantvekt og at grafen ikke inneholder negative sykler)

Bellman-Ford's algoritme

Tar utgangspunkt i en startnode.
Slakker (Relax) alle kanter i grafen, gjentar dette $|V|$ ganger.
(Forutsetter at grafen ikke inneholder negative sykler)

Bredde-først søk (BFS)

Tar utgangspunkt i en startnode.
Traverserer grafen etter FIFO-prinsippet.
(Forutsetter at alle kanter har lik, ikke-negativ vekt)

Alle til alle

Floyd-Warshalls algoritme

Sjekker for ethvert par med noder (**u**, **v**) om det er **lønnsomt** å gå innom node **k** på veien fra **u** til **v**.

Transitive closure

Sjekker for ethvert par med noder (**u**, **v**) om det er **mulig** å gå innom node **k** på veien fra **u** til **v**. Returner en *reachability matrix* som angir hvilke noder som kan nåes fra de ulike nodene i grafen.
Dette svarer til en binær implementasjon av **Floyd-Warshalls** algoritme.

Johnson's

Benytter **Bellman-Ford** til å gjøre en revekting av kantvektene i grafen.
Kjører så **Dijkstra's algoritme** fra hver node i grafen.