

Grafteori	Niveau 2	LÆRER
-----------	----------	-------

Den korteste vej – Eller fornøjelsen ved at følge en opskrift

Introduktion

Forløbet handler om, hvordan man kan bestemme den korteste rute i en vægtet graf. Den vægtede graf kan opfattes som en repræsentation af data om afstande mellem visse punkter. Fra Niveau 1 har eleverne stiftet bekendtskab med en sådan repræsentation. Det undersøgelsesbaserede aspekt består i elevernes udforskning af Dijkstras algoritme i forskellige implementeringer.

Udfoldende spørgsmål

Google Maps omfatter et gigantisk vejnet. Hvordan bestemmer Google Maps den korteste vej (eller hurtigste) i selv meget komplicerede tilfælde?

Forudsætninger

Erfaring med at bestemme afstande langs veje på kort (som fra niveau 1). Kendskab til begreberne graf, kanter, knuder, kantvægte, sti. Kendskab til sammenhæng mellem afstandsdata mellem punkter og vægtet graf. Tilstrækkelige CAS-kompetencer til at kunne implementere en grafpakke og følge simpel syntaks for nogle CAS-rutiner ud fra pdf-udskrifter af eksemplariske Maple-worksheet (vedlagt de konkrete tilfælde).

Tilsigtede læringsmål

At kunne redegøre for, hvad det vil sige at følge den konkrete algoritme (Dijkstra). At kunne anvende Dijkstras algoritme med blyant og papir for små vægtede grafer. At kunne redegøre for Dijkstras grundlæggende idé i forbindelse med udførelse af Dijkstras algoritme på konkret graf. At kunne lave en trin for trin illustration af Dijkstras algoritme ved hjælp af gentagne kald af Dijkstra applikation i et CAS-værktøj (her Maple) i konkrete tilfælde.

Forslag til hjælpemidler

Blyant (flere farver) og papir. Tavle og farvekridt. CAS-program med Dijkstra applikation, fx Maple.

Andet materiale

Udskrifter (pdf) af grafer, CAS-kommandoer, m.m. (Eleverne skal ikke have direkte adgang til CAS-filer, men selv foretage de nødvendige indtastninger)

Fase 1

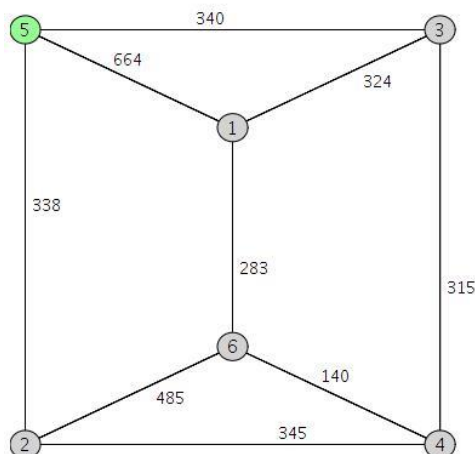
Eleverne præsenteres for en vægtet graf (se slides) og stifter bekendtskab med Dijkstras algoritme, version1 – både sprogligt og som rutediagram. Deres opgave er at finde korteste sti fra hjørnet S til hvert af grafens hjørner vha. Dijkstras algoritme. Dette kræver løbende opdateringer, fx vha. tabeller om opnåede korteste afstande undervejs, eller ved at tegne stierne ind på grafen efterhånden som algoritmen bestemmer dem. Disse opdateringer er ikke beskrevet i version 1. Meningen er, at eleverne selv skal opdage behovet for opdateringer ved at gå i gang med arbejdet. De kan derfor næppe gennemføre Dijkstras algoritme i denne fase.

Læreren starter med at gennemgå Dijkstras algoritme og sikrer sig, at der ikke er sproglige vanskeligheder, og at rutediagrammet er forståeligt. Derefter iagttages elevernes arbejde. Læreren noterer sig, hvilke vanskeligheder der opstår – især med opdateringsproblematikken. Fase 1 afsluttes, når det skønnes frugtbart i forhold til elevernes erkendelse af opdateringsbehovet.

Fase 2

Eleverne præsenterer deres hidtidige arbejde med henblik på afklaring af opdateringsproblematikken. Der udveksles erfaringer og gode råd.

Læreren introducerer metoder til at fastholde opdateringer. Det kan være en tabel over fremadskridende trin i algoritmen eller en tegneserie over, hvordan Dijkstra-stierne udvikler sig. Fx kan tabellen til opgaven på niveau 1



udformes således

\mathcal{D}	1	2	3	4	5	6
5	664	338	340	∞	0	∞
5, 2	664	338	340	683		823
5, 2, 3	664		340	655		823
5, 2, 3, 4	664			655		795
5, 2, 3, 4, 1	664					795
5, 2, 3, 4, 1, 6						795

De farvede felter angiver den korteste afstand.

Fase 3

Eleverne løser nu opgaven færdigt og præsenterer deres resultater: De korteste afstande og de stier, der er fundet. Herefter argumenteres for resultaterne, herunder hvordan stierne er fundet.

Læreren sammenfatter: Eleverne skal have fundet de samme korteste afstande, men de tilhørende Dijkstra-stier kan godt være forskellige. Hvis der er uenighed om afstandene, skal det løses.

Fase 4

Denne fase handler om at benytte kald af Dijkstras algoritme i et CAS-program, her illustreret ved Maple.

Eleverne får udleveret pdf-udskrift af Maple-dokument med kald af grafpakken og kommandoer for at oprette grafer, tegne grafer, fremhæve delgrafer, udføre Dijkstras algoritme, m.m. Ved hjælp af disse skal der produceres en tegneserie, der viser udviklingen af Dijkstra-stierne i den konkrete graf fra ppt-slides. I første omgang skal der lægges en produktionsplan for tegneserien. Hvordan skal input og output være for at få de enkelte billeder frem? Vent i første omgang med at taste ind.

Læreren observerer og noterer sig elevernes fremskridt. Elevernes produktionsplaner finpudses.

Fase 5

Eleverne færdiggør arbejdet med tegneserien og forbereder præsentation, herunder begrundelsen for deres produktionsplan.

Læreren sidder ved 'help-desk' for at hjælpe med konkrete CAS-problemer.

Fase 6

Eleverne præsenterer deres tegneserie og redegør for Dijkstras idé, fra billede til billede.

Læreren samler pointer op og formulerer den indvundne indsigt. Er de tilsigtede læringsmål nået, herunder forståelse af algoritmebegrebet? Nogle forhold som endnu er uafklarede (som kan tages op på Niveau 3) præciseres, så vidt muligt på baggrund af input fra eleverne – først og fremmest, om Dijkstras algoritme faktisk finder, hvad den lover (den korteste vej). Andre forhold kunne være: Hvordan beregnes opdateringer af afstande effektivt? Hvordan holder computeren rede på Dijkstra-stierne? Er Dijkstras algoritme effektiv for store vægtede grafer? Fx har USA et netværk af 6,4 millioner km offentlige veje!