Den korteste vej. Eller fornøjelsen ved at være effektiv.

Diskret matematik, niveau 3

Fase 1

Dijkstras algoritme, fuld version (1)

Input: En vægtet graf, G og et starthjørne, u. Vægten af kanten xy benævnes $w(xy)$; vi sætter $w(xy) = +\infty$, hvis xy ikke er en kant.

Idé: At vedligeholde mængden af hjørner, S, hvortil man kender en korteste vej fra u, og trin for trin at udvide S, til den omfatter alle hjørner i G. For at gøre dette holdes regnskab med en foreløbig afstand, $t(z)$ fra u til ethvert $z \notin S$: vi kan tænke på $t(z)$ som en seddel på hjørnet z, hvor vi hele tiden noterer længden af den hidtil korteste $u, z -$ sti.

Dijkstras algoritme, fuld version (2)

Initialisering:

Vi starter med $S = \{u\}$; $t(u) = 0$; $t(z) = w(uz)$ for $z \neq u$.

Iteration:

a. Vælg hjørne $v \notin S$ så $t(v) = \min_{z \notin S} t(z)$.
b. Tilføj v til S.
c. For hvert hjørne $z \notin S$ opdateres $t(z)$ til $\min\{t(z), t(v) + w(vz)\}$.

Iterationen fortsætter indtil $S = V(G)$. Herefter sættes $\mathrm{d}(u, v) = t(v)$ for alle $v \in V(G)$.

Fase 2

Rollespil af algoritmen (1)

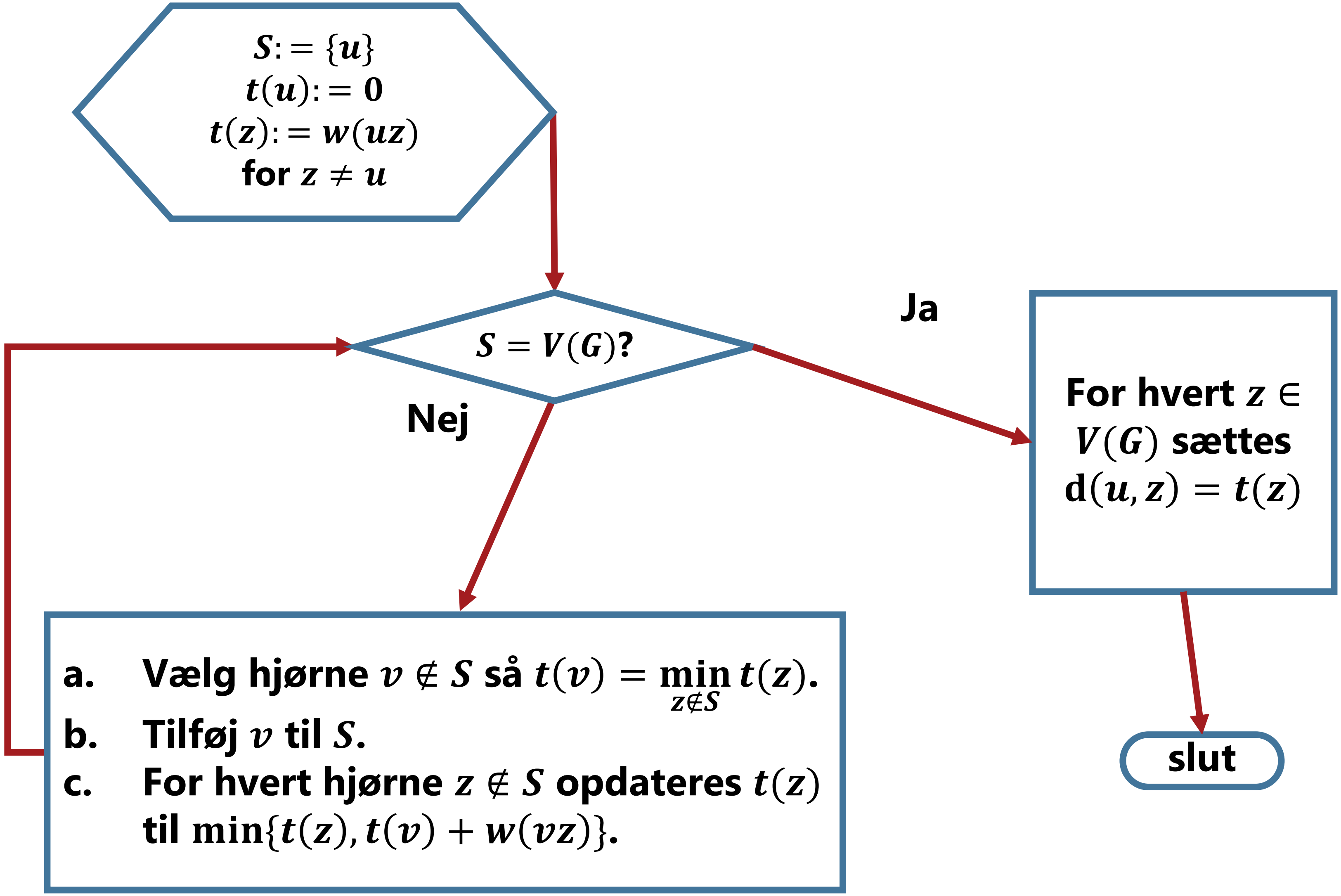
Algoritmen består i en række instruktioner til computeren. At udføre disse instruktioner kan opfattes som en rolle. Vi illustrerer dette ved at I laver et rollespil.

- Hver instruktion udføres af en elev, fx opdatering af $t(z)$ for et bestemt hjørne eller valg af nyt hjørne.
- Rolleindehaveren skal ikke foretage sig andet end hvad der står i rollen (forestil dig at du er en celle i computeren, der blot gør hvad den er programmeret til.)
- Rutediagrammet opfattes som manuskriptet til rollespillet.

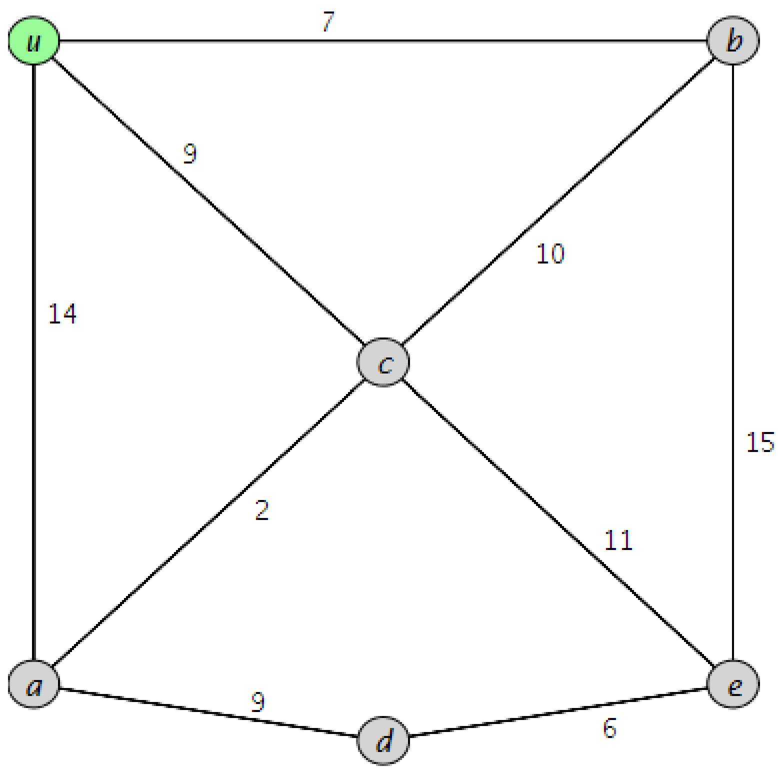
Rollespil af algoritmen (2)

- Fordeling af roller så man får 1 – 3 ″elevcomputere″.
- Gennemførelse af rollespillet vha. rutediagrammet på den anførte graf.
- ″Elevcomputerens″ resultater sammenlignes med en ægte computerberegning.
- Evaluering af rollespillet

Dijkstras algoritme, rutediagram



Vægtet graf til rollespil med Dijkstras algoritme



Fase 3

Bevis for at Dijkstras algoritme virker

Beviset er udleveret på pdf-ark.

Læsestrategi: Stop-and-go

Man læser ord for ord – langsomt.

- Ved hvert ord spørger man 'Er jeg med hertil? indtil man må svare 'Nej'.
    - Kan problemet løses her og nu? Notér løsningen!
    - Ellers notér hvori problemet består
- Læs videre til næste stop.

Dette gentages med nogle fokuspunkter, som læreren fastlægger. På denne måde kommer man så langt i sin forståelse, som det er muligt for nuværende.

Opbygning af matematiske sætninger

Matematiske sætninger kan altid omformes på en bestemt skabelon-agtig måde. Det er denne skabelon der ligger til grund for et bevis for sætningen.

- **Sætningens scenario:** Utvetydig beskrivelse af de objekter og definitioner, som fastlægger hvad sætningen handler om.
- **Forudsætninger** som vi antager om scenariet
- Den **ønskede konklusion**, hvis antagelserne er opfyldte.
- Bevis: **Logisk deduktion** *forudsætninger* $\Rightarrow$ *ønsket konklusion*

Fase 4

Hvor effektiv er Dijkstras algoritme?

Større grafer tager længere beregningstid. Vi angiver størrelse ved antallet af hjørner, n.

Algoritmens effektivitet skønnes ved at opregne hvor mange instruktioner computeren skal udføre i løbet af algoritmen. Instruktioner er (i vores tilfælde)

- Tildeling (fx tilføjelse af et nyt hjørne til S)
- Addition
- Sammenligning af 2 tal (fx bestem det mindste)

Giv et skøn af hvor mange instruktioner Dijkstras algoritme behøver til en graf med n hjørner.