Minimalt spännande träd i viktad graf

Ett spännande träd för en graf G är en delgraf till G som är ett träd (sammanhängande, har inga cykler) och innehåller alla hörn i G.

Vikten för ett spännande träd är summan av dom ingående kanternas vikter.

Prims algoritm:

Indata: Graf $G=\langle V,E\rangle$, kantvikter $f: E\to\mathbb{R}$, starthörn s

Utdata: Ett minimalt spännande träd för G lagrat med faderspekare $\pi(u)$

Prim(V, E, f, s) =

```
Q ← V
för varje u ∈ Q:
    key[u] ← ∞
key[s] ← 0      (Q är en heap med s överst)
π[s] ← NIL
while Q ≠ ∅ do
    u ← HeapExtractMin(Q)
    för varje granne v till u:
        if v ∈ Q and f(u,v) < key(v) then
            π[v] ← u
            key[v] ← f(u,v)    (här måste v flyttas i heapen)
```

Tidskomplexitet: $\mathcal{O}(|V|\log|V| + |E|\log|V|) = \mathcal{O}(|E|\log|V|)$

Minimalt spännande träd, Kruskal

Kruskals algoritm:

Indata: Graf $G=\langle V,E\rangle$, kantvikter $f: E\to\mathbb{R}$

Utdata: Ett minimalt spännande träd för G lagrat som en kantmängd $A\subseteq E$.

Kruskal(V, E, f) =

```
A ← ∅
för varje u ∈ V
    MakeSet(u)
sortera kanterna i E efter stigande vikt
för varje kant (u,v) ∈ E i stigandeviktsordning:
    if FindSet(u) ≠ FindSet(v) then
        A ← A ∪ {(u,v)}
        Union(u,v)
returnera A
```

Komplexitetsanalys:

MakeSet(u) tar tid $\mathcal{O}(1)$

FindSet och Union tar tid $\mathcal{O}(\log|V|)$

Sorteringen av E tar tid $\mathcal{O}(|E|\log|E|)$

Totalt: $\mathcal{O}(|V|\cdot 1 + |E|\log|E| + |E|\log|V|) = \mathcal{O}(|E|\log|E|)$ om grafen är sammanhängande

Korrekthet för Prim och Kruskal

Idé: Visa att varje kant som läggs till i algoritmen är säker, dvs ingår i något MST.

Definitioner:

- Ett **snitt** (cut) är en delning av V i S och $V-S$.
- En kant **korsar** snittet om ena änden $\in S$ och andra $V-S$.

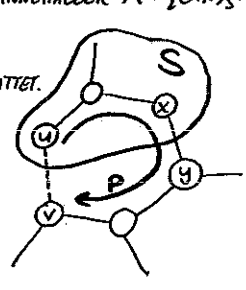
Sats: Givet $G=\langle V,E\rangle$, $f: E\to\mathbb{R}$, $A\subseteq E$, $S\subseteq V$. Om

- det finns ett MST som innehåller A,
- ingen kant i A korsar snittet $(S, V-S)$,
- (u,v) är det lättaste kant som korsar snittet

så är (u,v) säker att lägga till, dvs det finns ett MST som innehåller $A\cup\{(u,v)\}$.

Bevis: Låt T vara MST som innehåller A men inte (u,v). Konstruera T' som är ett MST och innehåller $A\cup\{(u,v)\}$:

T innehåller stig p mellan u och v.
Det finns kant (x,y) i p som korsar snittet.
Låt $T' = T\cup\{(u,v)\} - \{(x,y)\}$.



T' är uppenbart ett spännande träd

(u,v) är den lättaste korsande kanten

$\Rightarrow f(u,v)\le f(x,y) \Rightarrow |T'|\le|T|$

$\Rightarrow T'$ är MST.

ALGORITM FÖR GRAFPROBLEMET "KORTASTE STIG"

DIJKSTRAS ALGORITM:

INDATA: $G=\langle V,E\rangle$, $f:E\to\mathbb{N}$, $s\in V$, $t\in V$

UTDATA: LÄNGDEN AV DEN KORTASTE STIGEN I G FRÅN s TILL t

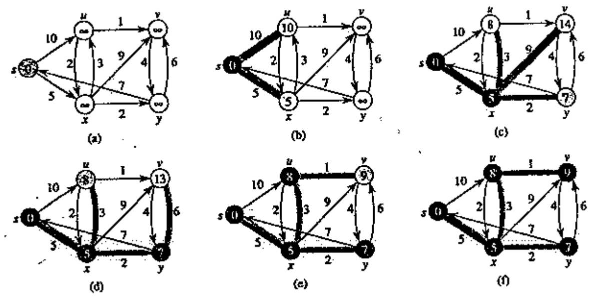
MÄRK VARJE HÖRN MED DET HITTILLS KORTASTE KÄNDA AVSTÅNDET FRÅN S.

UPPRÄTTA EN MÄNGD S MED DOM HÖRN TILL VILKA DEN OPTIMALA KORTASTE STIGEN ÄR KÄND.

ALGORITM:

1. FÖR VARJE HÖRN $u\in V$:
   OM $(s,u)\in E$ MÄRK u MED $f(s,u)$
   ANNARS MÄRK u MED ∞
2. MÄRK s MED 0 OCH LÅT $S=\{s\}$
3. SÅ LÄNGE $t\notin S$:
   UTVIDGA S MED DET HÖRN SOM ÄR MÄRKT MED DET KORTASTE AVSTÅNDET OCH UPPDATERA HÖRNMÄRKNINGEN
4. RETURNERA AVSTÅNDET SOM t ÄR MÄRKT MED

EXEMPEL PÅ DIJKSTRAS ALGORITM:



ANALYS:

S UTVIDGAS $|V|$ GÅNGER (HÖGST).

VID VARJE UTVIDGNING LETAR MAN UPP DET HÖRN SOM ÄR MÄRKT MED KORTASTE AVSTÅNDET: $O(|V|)$

UPPDATERING AV HÖRNMÄRKNINGEN GÖRS HÖGST EN GÅNG FÖR VARJE KANT I GRAFEN: $O(|E|)$

INITIERING AV S OCH MÄRKNINGEN TAR TID $O(|V|+|E|)$

TOTALT: $O(|V|^2+|E|+|V|+|E|) = O(|V|^2)$

(EFTERSOM $|E|\in O(|V|^2)$)

KORREKTHET FÖR DIJKSTRAS ALGORITM

LÅT $\delta(s,v)$ VARA DET KORTASTE AVSTÅNDET FRÅN s TILL v.
LÅT $d[v]$ VARA HÖRNET v-S MÄRKNING I ETT LÄGE I ALGORITMEN.

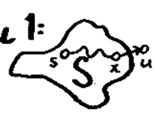
BEVISSKISS:

NOTERA ATT $d[v]\geq\delta(s,v)$ ALLTID GÄLLER FÖR ALLA HÖRN.

INDUKTION ÖVER S:

BASFALL: $S=\{s\}$, $d[s]=0$, $\delta(s,s)=0$ **OK!**

INDUKTIONSSTEG: VISA ATT OM $d[v]=\delta(s,v)$ FÖR ALLA $v\in S$ NÄR u JUST SKA LÄGGAS TILL S SÅ ÄR $d[u]=\delta(s,u)$.

FALL 1:  KORTASTE STIGEN FRÅN s TILL u GÅR HELT INUTI S UTOM SISTA KANTEN (x,u).

ANTAGANDET $\Rightarrow d[x]=\delta(s,x)$
ALGORITMEN SATTE $d[u]=d[x]+f(x,u)=\delta(s,x)+f(x,u)=\delta(s,u)$ **OK!**

FALL 2:  LÅT y VARA FÖRSTA HÖRNET UTANFÖR S I KORTASTE STIGEN FRÅN s TILL u.

FALL 1 $\Rightarrow d[y]=\delta(s,y)\leq\delta(s,u)$
ALGORITMEN LÄGGER TILL u FÖRE $y \Rightarrow d[u]\leq d[y]$
VI HAR NU: $d[y]\leq\delta(s,u)\leq d[u]\leq d[y]$
$\Rightarrow d[y]=\delta(s,u)=d[u]$ **OK!**

ALLA HÖRN SOM KAN NÅS FRÅN s KOMMER MED I S. ÖVRIGA HAR $d[v]=\infty$