

Föreläsning 14

Dijkstra's kortaste väg algoritm.

Matrisrepresentationer av grafer

Grafisomorfier

Invarianter

En *viktad graf* är en graf med ett tal (vikt) $w(e)$ associerat till varje kant.

(En fktn $w : E \rightarrow \mathbb{R}$)

Längden till en väg

$p = v_0, e_1, v_1, e_2, v_2, e_3 \dots, e_k, v_k$

i en viktad graf är

$$l(p) = \sum_{i=1}^k w(e_i)$$

Dijkstras kortaste väg algoritm är en algoritm för att bestämma längden av kortaste vägen mellan två hörn i en viktad graf med positiva vikter.

Invärden: sammanhängande viktad graf $G = (V, E)$ med hörn a, z

Utvärde: Längden $L(z)$ av kortaste vägen mellan a och z .

I pseudokod:

Dijkstra(w, a, z, L) {

$L(a) := 0$

$L(x) := \infty$ för alla hörn $x \neq a$

$T :=$ mängden av alla hörn.

 // T kommer alltid vara de hörn för vilka vi ännu inte bestämt kortaste avstånd till

a .

while($z \in T$) {

 välj $v \in T$ så att $L(v) = \min_{x \in T} L(x)$

$T := T \setminus \{v\}$

 för varje $x \in T$ granne med v

$L(x) := \min\{L(x), L(v) + w(v, x)\}$

 }

 return $L(z)$

}

Varje gång whileloopen körs plockas ett hörn v bort ur T . Första gången är det a som plockas bort. För det element som plockas bort så gäller att $L(v)$ är kortaste avstånd från v till a .

Så småningom måste det vara z som plockas bort, då är algoritmen klar .

För beviset att Dijkstras algoritm verkligen beräknar längden av kortaste vägen mellan a och z se kursboken.

För en enkel sammanhängande, viktad graf med n hörn har Dijkstra's algoritm en worst-case tid på $\Theta(n^2)$.

Man kan också mha algoritmen spåra vilken väg som är den kortaste mellan a och z . Inte bara längden på den kortaste vägen.

Grannmatrisen till en graf $G = (V, E)$, givet en ordning $v_1, v_2, \dots, v_n$ av hörnen, är en (symmetrisk) $n \times n$ matris $A = (a_{ij})$ definierad genom

a_{ij} = antalet kanter mellan v_i och v_j om $i \neq j$.

a_{ii} = 2 ggr antalet loopar på i .

Insidensmatrisen till en graf $G = (V, E)$, givet en ordning $v_1, v_2, \dots, v_n$ av hörnen, och en ordning $e_1, \dots, e_m$ av kanterna är en $n \times m$ matris $B = (b_{ij})$ definierad genom

$b_{ij} = 1$ om v_i är en av ändpunkterna till e_j

$b_{ij} = 0$ annars

Definition Två grafer $G_1 = (V_1, E_1)$ och $G_2 = (V_2, E_2)$ är isomorfa om:

Det finns bijektioner $f : V_1 \rightarrow V_2$ och $g : E_1 \rightarrow E_2$ så att

om $e \in E_1$ har ändpunkter v och w så har $g(e)$ ändpunkterna $f(v)$ och $f(w)$.

Sats Om två grafer G_1 och G_2 är isomorfa så finns det någon ordning av deras hörn för vilken de har samma grannmatris.

Motivation: Om A är G_1 's grannmatris m a p $v_1, v_2, \dots, v_n$ så har G_2 A som grannmatris m a p ordningen $f(v_1), f(v_2), \dots, f(v_n)$.

Omvänt om två grafer har samma matris kan man definiera en isomorfi utifrån detta.

För att visa att två grafer *inte* är isomorfa kan man använda sig av så kallade *invarianter*.

Detta är egenskaper så att om G_1 och G_2 är isomorfa så har de antingen båda denna egenskap eller så har de den inte.

OBS Även om G_1 och G_2 inte är isomorfa så kan de ha samma invarianta egenskap.

Exempel på invarianter

' G har n kanter'

' G har m hörn'

' G har ett hörn av grad k '

' G har en enkel cykel av längd l '

De första två följer omedelbart av definitionen. (Bijektion mellan E_1 och E_2 och mellan V_1 och V_2).

Den tredje: Anta G_1, G_2 isomorfa. G_1 har ett hörn v så att $\delta(v) = k$. Då finns kanter $e_1, \dots, e_k$ med v som ändpunkt. Då är $g(e_1), \dots, g(e_k)$, precis de kanter som har $f(v)$ som ändpunkt. Dvs $\delta(f(v)) = k$

Den fjärde. Om $v_0, v_1, \dots, v_k$ är en enkel cykel av längd k i G_1 så är $f(v_0), f(v_1), \dots, f(v_k)$ en enkel cykel av längd k i G_2 .

Exempel: Ej isomorfa, olika många hörn.

Exempel: Ej isomorfa, olika många kanter

Exempel: Ej isomorfa, ena grafen har hörn med grad 3, den andra inte

Exempel: Ej isomorfa, ena grafen har 3-cykel den andra inte.