Índex

Aspectes computacionals

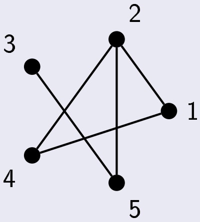
Objectiu

Representar i treballar computacionalment amb grafs

Representació: Diccionari de vèrtexos adjacents

Per a cada vèrtex, guardar la llista de vèrtexos adjacents a ell

Exemple



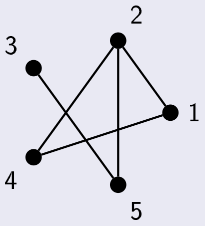
v	$\mathcal{A}(v)$
1	2,4
2	1,4,5
3	5
4	1,2
5	2,3

Representació: Matriu d'adjacència

Enumerar vèrtexos $v_1, \ldots, v_n$ i construir matriu

$$A = (a_{i,j}), \qquad a_{i,j} = \begin{cases} 1 & \text{si } v_i v_j \in E \\ 0 & \text{altrament.} \end{cases}$$

Exemple



$$\begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{pmatrix}$$

Proposició

Sigui $a_{i,j}^{(k)}$ l'entrada (i,j) de la matriu A^k. Aleshores $a_{i,j}^{(k)}$ és igual al nombre de recorreguts $v_i \rightsquigarrow v_j$ de longitud k

Demostració

- Cert per definició per a $k = 1$
- Per a $k > 1$: Per a cada $v_i, v_j \in V$, bijecció entre:
  - Camins $v_i \rightsquigarrow v_j$ de long. k
  - Parells $(v_i \rightsquigarrow v_l, v_l v_j \in E)$ amb $u \rightsquigarrow w$ de long. $k-1$

Per H.I., núm. camins: $\sum_j a_{i,l}^{(k-1)} a_{l,j}^{(1)} = a_{i,j}^{(k)}$ □

Observació

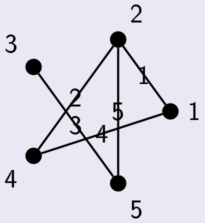
Amb grafs dirigits funciona igual

Representació: Matriu d'incidència (cas no dirigit)

Enumerar vèrtexos $v_1, \ldots, v_n$ i arestes $e_1, \ldots, e_m$ i construir matriu

$$B = (b_{i,j}), \qquad a_{i,j} = \begin{cases} 1 & \text{si } v_i \text{ incident amb } e_j \\ 0 & \text{altrament.} \end{cases}$$

Exemple



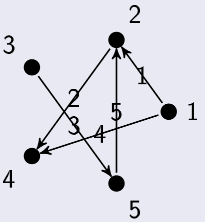
$$\begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix}$$

Representació: Matriu d'incidència (cas dirigit)

Enumerar vèrtexos $v_1, \ldots, v_n$ i arcs $e_1, \ldots, e_m$ i construir matriu

$$B = (b_{i,j}), \qquad a_{i,j} = \begin{cases} 1 & \text{si } v_i \text{ v. inicial de } e_j \\ -1 & \text{si } v_i \text{ v. final de } e_j \\ 0 & \text{altrament.} \end{cases}$$

Exemple



$$\begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ -1 & 1 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & 1 \end{pmatrix}$$

Teorema

G graf, A mat. d'adjacència, B matriu d'incidència, D matriu diagonal amb $d_i = d(v_i)$. Aleshores:

$$B \cdot B^t = A + D$$

Demostració

$V = \{v_1, \ldots, v_n\}$, $E = \{e_1, \ldots, e_m\}$.

- Si $i \neq j$, $(B \cdot B^t)_{i,j} = \sum_k b_{i,k} b_{j,k}$. Cada sumand: 1 si $e_k = v_i v_j$; 0 altrament. La suma és $a_{i,j}$.
- Si $i = j$, $(B \cdot B^t)_{i,i} = \sum_k b_{i,k} b_{i,k}$. Cada sumand: 1 si e_k té extrem v_i; 0 altrament. La suma és d_i.

Algorismes sobre grafs

Els llenguatges d'alt nivell implementen:

- Grafs i digrafs com a tipus de dades
- Mètodes per introduir grafs
- Mètodes per a tractar grafs:
  - Accés a vèrtexos, arestes
  - Accés a veïns d'un vèrtex
  - ...
- Algorismes específics

Algorisme de Dijkstra

Donat un graf (dirigit) amb pesos als arcs, trobar el camí de pes mínim entre un vèrtex donat i els altres (Demo)

Dades: Un graf G i un node orígen u.

$\mathrm{NoOpt} := V$;
per a *tot node* v *diferent de* u **fer**
    $\mathrm{dist}(v) := \infty$;
    $\mathrm{pred}(v) := \mathrm{nodef}$;
fi
$\mathrm{dist}(u) := 0$;
mentre $\mathrm{NoOpt} \neq \emptyset$ **fer**
    Sigui $v \in \mathrm{NoOpt}$ amb $\mathrm{dist}(v)$ mínim;
    $\mathrm{NoOpt} := \mathrm{NoOpt} \setminus \{v\}$;
    per a *tot node* v' *de* NoOpt *adjacent a* v **fer**
        si $\mathrm{dist}(v) + w(vv') < \mathrm{dist}(v')$ **aleshores**
            $\mathrm{dist}(v') = \mathrm{dist}(v) + w(vv')$;
            $\mathrm{pred}(v') = v$;
        fi
    fi
fi

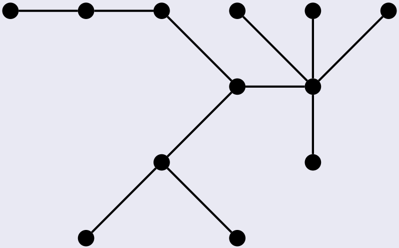
Sortida: Taula amb $\mathrm{dist}(v)$ i $\mathrm{pred}(v)$

Arbres no arrelats

Arbres (no arrelats)

Un *arbre* és: graf connex acíclic

Exemple



Proposició (caracteritzacions d'arbres)

En tot graf, són equivalents:

1. G és connex i acíclic.
2. Tot parell de vèrtexos de G està unit per un únic camí.
3. G és connex i, si el seu ordre és n, la seva mida és $n-1$.
4. G és connex, però $G-e$ és no connex per a tota aresta $e \in E(G)$.
5. G és acíclic, però $G+uv$ conté un cicle per a tot parell u, v de vèrtexos independents.

Demostració

- ($1 \implies 2$) El camí existeix per ser G connex. Si hi ha múltiples camins, obtenim cicle.
- ($2 \implies 1$) G és connex per l'existència de camins. Si hi ha cicles, obtenim múltiples camins.
- ($1 \implies 3$) Fem inducció sobre $|V| = n$. Per a $n = 1$ el resultat és trivial. Sigui G un graf connex acíclic amb $n + 1$ vèrtexos, sigui e una aresta qualsevol i G/e el graf amb n vèrtexos obtingut per contracció de e. G/e és connex i acíclic; per hipòtesi d'inducció G/e té $n - 1$ arestes, d'on G en té n.
- ($3 \implies 1$) Fem inducció sobre $|V| = n$. Per a $n = 1$ el resultat és trivial. Sigui G amb $n + 1$ vèrtexos i n arestes, i sigui e una aresta qualsevol. G/e té n vèrtexos i $n - 1$ arestes, d'on és connex i acíclic. Ara G és connex i acíclic (per ser-ho G/e).

Demostració

- $(1 \implies 4)$ Suposem que $G - e$ és connex per a certa aresta $e = uv$. Un camí $u \rightsquigarrow v$ a $G - e$ concatenat amb e dóna cicle. Contradicció.
- $(4 \implies 1)$ Suposem que G té cicle, i sigui e aresta del cicle. El graf $G - e$ és connex. Contradicció.
- $(1 \implies 5)$ Siguin u, v independents; considerem $u \rightsquigarrow v$ únic camí de u a v. Concatenant amb $e = vu$ trobem cicle.
- Suposem G no connex, i u i v de components diferents. $G + uv$ no pot contenir cicles si G no té cicles. Contradicció.

Boscos

Un *bosc* és un graf acíclic (components connexos són arbres)

Arbres generadors

Un *arbre generador* d'un graf connex és subgraf generador (conté tots els vèrtexos) que és arbre.

Proposició

Tot graf connex té arbre generador

Demostració.

Comencem amb $W = \{u_0\}$ un vèrtex qualsevol i $F = \{\}$. $T = (W, F)$ és un arbre. Per a cada $k = 2, \ldots, |V|$ fem:

- Escollim $e = uv$ aresta unint vèrtex u de W amb vèrtex v de $V \setminus W$ (existeix per connexitat)
- $W := W \sqcup \{v\}$, $F := F \sqcup e$, $T := (W, F)$

Al final F és subgraf de G amb n vèrtexos i $n - 1$ arestes. $\square$

Arbres generadors minimals

- Un *graf amb pesos a les arestes* és un graf amb funció $w : E \to \mathbb{R}^+$.
- El *pes* d'un subgraf és la suma dels pesos de les arestes que conté.
- Un *arbre generador minimal* és un arbre generador de pes minimal.

Algoritme de Prim

Donat un graf amb pesos a les arestes, trobar un arbre generador minimal. (Demo)

Dades: Un graf G amb pesos a les arestes.
Sigui $e = uv$ una aresta de G de pes minimal; $V_1 := \{u, v\}$; $T := (V_1, \{e\})$;
per a $k = 2, \ldots, |V| - 1$ **fer**
  Sigui $e_k = (u_k v_k)$ de pes mínimal t.q. $u_k \in V_{k-1}$, $v_k \notin V_{k-1}$;
  Fem $V_k := V_{k-1} \cup \{v_k\}$;
  Fem $T := T + e_k$;
fi
Sortida: T

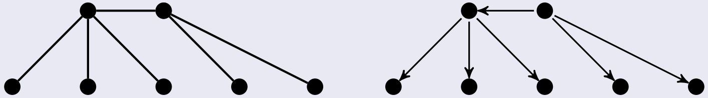
Arbres arrelats

Arbres arrelats

Un *arbre arrelat* és:

- Arbre amb node distingit (arrel)
- Digraf amb:
  - únic vèrtex r amb $d_e(r) = 0$
  - Per a tot $u \in V$ $\exists!$ camí $r \rightsquigarrow u$

Exemple



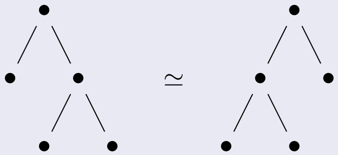
Notacions

- Si uv és arc: u és *pare* de v, v és *fill* de u
- Arrel: únic node sense pare
- *Fulla*: node sense fills
- *Node interior*: node amb pare (únic) i fills
- *Node elemental*: node amb únic fill
- *Descendència* de u: Nodes accessibles des de u
- *Ascendència* de u: Nodes des dels que es pot accedir a u

Arbres binaris

Tot node no fulla té 2 fills (indistingibles)

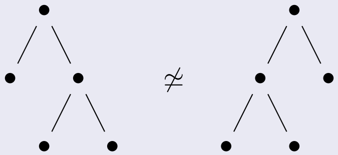
Exemple



Arbres ordenats

Arbre on es fixa ordenació dels fills dels nodes interiors

Exemple



Arbres ordenats binaris

Tot node no fulla té dos fills distingibles (esq./dreta)

Recompte d'arbres

Nombre d'arbres amb n fulles:

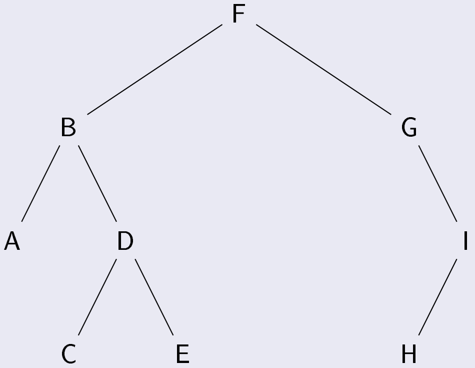
n	SNE	Bin.	Bin. Ord.
2	1	1	1
3	2	1	2
4	5	2	5
5	12	3	14
6	33	6	42
7	90	11	132
8	261	23	429
9	766	46	1430
10	2312	98	4862

- SNE: Sense nodes elementals
- Bin.: Binaris
- Bin. Ord.: Binaris ordenats

Recorreguts

- Recorregut en *preordre* de T:
  1. Visitar l'arrel de T.
  2. Recorre en preordre el subarbre de l'esquerra de T.
  3. Recorre en preordre el subarbre de la dreta de T.
- Recorregut en *inordre* de T:
  1. Recorre en inordre el subarbre de l'esquerra de T.
  2. Visitar l'arrel de T.
  3. Recorre en inordre el subarbre de la dreta de T.
- Recorregut en *postordre* de T:
  1. Recorre en postordre el subarbre de l'esquerra de T.
  2. Recorre en postordre el subarbre de la dreta de T.
  3. Visitar l'arrel de T.

Exemple



- Preordre: F, B, A, D, C, E, G, I, H.
- Inordre: A, B, C, D, E, F, G, H, I.
- Postordre: A, C, E, D, B, H, I, G, F.