

ESTRUCTURES DE DADES DELS GRAFS

En els grafs, normalment s'ha de tractar una gran quantitat d'informació: nodes, arestes, pesos, sentits... Tota aquesta informació no és difícil de gestionar si el graf que s'estudia és petit, ja que fins i tot es pot dibuixar. El problema sorgeix quan el graf en qüestió és més gran, com per exemple podria ser una xarxa de clavagueram o de metro. Llavors la informació a tractar és molta, i és convenient organitzar-la en estructures de dades. Les estructures de dades solen ser utilitzades en la programació, però en aquest cas sol ser beneficiós tenir-ne fins i tot quan no es treballa amb informàtica.

Matrius

Matriu d'adjacència

La matriu d'adjacència d'un graf és una matriu quadrada que conté informació respecte el nombre d'arestes que uneixen dos nodes, i s'organitza de la següent manera:

Si G és un graf amb n nodes, $A(G) = (a_{i,j})_{i,j=1,\dots,n}$ és la seva matriu d'adjacència de $n \times n$, on $a_{i,j}$ correspon al nombre d'arestes que uneixen els nodes i, j , contant com a 2 els llaços, que sempre es trobaràn a la diagonal. La matriu serà simètrica si el graf ho és, i podrem conèixer el grau d'un node i fent el sumatori de les caselles de la i -èsima fila. A vegades, quan s'utilitzen grafs ponderats, les matrius d'adjacència s'omplen amb els pesos de les arestes, per no haver d'utilitzar altres estructures per emmagatzemar la resta d'informació. En aquest tipus de grafs, els llaços no necessàriament valdràn 2, però es podran diferenciar perquè estaràn a la diagonal. En aquest cas, el problema serà que no es podran representar arestes múltiples amb pesos diferents. Tot i això, en termes de programació, utilitzar aquesta estructura de dades (sobretot en grafs grans) no és el més eficient, ja que creix molt ràpidament. La matriu d'un graf de n nodes té n^2 espais, i en grafs poc densos, la majoria d'espais estan ocupats per 0, de tal manera que s'està utilitzant molta memòria que no conté informació útil. De la mateixa manera, si afegim un nou node al graf, que serà l' n -èssim, s'hauràn d'afegir $2n - 1$ espais a la matriu.

Matriu d'incidència

La matriu d'incidència d'un graf G sense llaços, $I(G) = (b_{i,j})_{i=1,\dots,v,j=1,\dots,\alpha}$, on $v = |V|$ i és la matriu binària $v \times \alpha$ on $b_{i,j}$ indica si la aresta j és incident al node i .

(afegir exemple de graf amb les seves dues matrius)

Llistes d'adjacència

Aquesta estructura de dades és la més utilitzada per tractar grafs, ja que ocupa menys memòria i només inclou l'informació necessària. Les llistes d'adjacència consisteixen en un conjunt de n llistes. Cada llista correspon a un node del graf, i conté els nodes al quals és adjacent. En informàtica s'acostumen a fer mitjançant apuntadors, de tal manera que, a partir de cada element de la llista, es pugui accedir a la seva pròpia llista, seguint així molt més senzill fer iteracions. Amb l'esquema següent es pot veure més clarament:

(afegir esquema de les llistes d'adjacència)

1 Camins i algorismes

Sovint, quan utilitzem un graf per modelitzar quelcom, ens interessa poder-hi fer algunes operacions. Podem, per exemple, voler trobar un camí entre dos punts, recórrer el graf sencer o trobar el camí més curt per anar d'un vèrtex a un altre. Per aquest motiu utilitzem els camins, que trobarem o generarem mitjançant diversos algorismes. En aquesta secció mostraré diverses maneres de recórrer un graf, torbant la manera més eficient per a cada cas.