

PRÀCTICA 2: Representació en llistes d'adjacències. Recorreguts jeràrquics i camins minimal

Considera la solució/projecte en Visual C++ 2017 Community que consta en `GPr2.zip` amb els fitxers de capçalera (*.h), els mòduls del programa (*.cpp) i els fitxers de dades de grafs `graph?.in` i de grafs ponderats `wgraph?.in` que l'acompanyen. Pots obrir-ho clicant el fitxer de solució `GPr2.sln`.

El mòdul `graph.cpp` conté:

- La funció que emplena les llistes d'adjacències d'un graf complet K_n amb el nombre de vèrtexs indicat.
- La funció que emplena les llistes d'adjacències d'un graf a partir de les dades següents d'un graf en un fitxer amb el format següent:
 - nombre de vèrtexs i arestes
 - parelles de vèrtexs que conformen les arestes
- La funció que envia a un *stream* de sortida informació diversa a partir de les llistes d'adjacències d'un graf:
 - nombre de vèrtexs i arestes
 - llistes d'adjacències dels vèrtexs
 - parelles de vèrtexs conformant les arestes
 - graus dels vèrtexs

El mòdul `wgraph.cpp` conté:

- La funció que emplena les llistes d'adjacències ponderades d'un graf complet WK_n amb el nombre de vèrtexs indicat ponderat amb pesos aleatoris fins a un pes màxim.
- La funció que emplena les llistes d'adjacències ponderades d'un graf a partir de les dades següents d'un graf en un fitxer amb el format següent:
 - nombre de vèrtexs i arestes
 - parelles de vèrtexs que conformen les arestes amb el pes corresponent
- La funció que envia a un *stream* de sortida informació diversa a partir de les llistes d'adjacències ponderades d'un graf:
 - nombre de vèrtexs i arestes
 - llistes d'adjacències ponderades dels vèrtexs
 - parelles de vèrtexs conformant les arestes amb el pes corresponent

El mòdul `graphTS.cpp` conté funcions de cerca jeràrquica (Tree Search) a partir de les llistes d'adjacències:

- la funció `DFS` fa la cerca en profunditat (Depth First Search) i retorna el nombre de components del graf,
- la funció `BFS` fa la cerca en extensió (Breadth First Search) i retorna el nombre de components del graf,
- la funció `Dijkstra` troba la distància de tots els vèrtexs a un vèrtex donat, entent la distància entre dos vèrtexs com la longitud mínima dels camins que els uneixen atenent al seu nombre d'arestes.

Els fitxers de capçalera `graph.h`, `wgraph.h` contenen els prototips de les diverses funcions i s'inclouen en els mòduls corresponents.

El programa principal que consta en el mòdul `GPr2.cpp`:

- Construeix en `K8` i `G0` les llistes d'adjacències de graf complet de 8 vèrtexs K_8 i del graf en `graph0.in`. A continuació, escriu la informació corresponent en `K8.out` i `graph0.out`, respectivament, incloent-hi el nombre de components trobades via `DFS` i `BFS` i les distàncies (en nombre d'arestes) al vèrtex 0.
- Construeix en `WK8` i `WG0` les llistes d'adjacència ponderades d'un graf complet de 8 vèrtexs ponderat amb pesos aleatoris fins a un pes màxim de 9, WK_8 i del graf ponderat corresponent al fitxer `wgraph0.in` (els pesos apareixen en la tercera columna de les arestes). A continuació, escriu la informació corresponent en `WK8.out` i `wgraph0.out`, respectivament.

Exercici 8 Amplia el programa per tal que tracti els grafs bipartits complets K_{n_1, n_2} , els grafs cicles C_n , els grafs estrelles S_n i els grafs rodes W_n , de la mateixa manera que els grafs complets K_n , sense i amb ponderació aleatòria.

Continua el programa principal:

- Construint en `(K5_3, K3_5)`, `C8`, `S8`, `W8` les llistes adjacències dels grafs $(K_{5,3}, K_{3,5})$, C_8 , S_8 , W_8 , i escrivint-ne la informació en `K5_3.out`, `C8.out`, `S8.out`, `W8.out`.
- Construint en `G1`, `G2`, `G3` les llistes d'adjacències dels grafs en `graph1.in`, `graph2.in`, `graph3.in` i escrivint-ne la informació en `graph1.out`, `graph2.out`, `graph3.out`.
- Construint en `(WK5_3, WK3_5)`, `WC8`, `WS8`, `WW8` les llistes adjacències ponderades dels grafs amb pesos aleatoris $(WK_{5,3}, WK_{3,5})$, WC_8 , WS_8 , WW_8 i escrivint-ne la informació en `WK5_3.out`, `WC8.out`, `WS8.out`, `WW8.out`.
- Construint en `WG1`, `WG2`, `WG3` les llistes d'adjacències ponderades dels grafs ponderats en `wgraph1.in`, `wgraph2.in`, `wgraph3.in` i escrivint-ne la informació en `wgraph1.out`, `wgraph2.out`, `wgraph3.out`.

Exercici 9 Completa el programa amb funcions `DFS_Trees` i `BFS_Trees` que guardin els arbres DFS i BFS corresponents als recorreguts jeràrquics dels diferents components i enviïn la informació següent a un *stream* de sortida:

- dels vèrtexs: l'identificador, l'índex en el recorregut, l'identificador del seu predecessor en l'arbre, si escau, i el seu nivell respecte a l'arrel;
- de les arestes, identificadors dels vèrtexs extrems i si són o no dels arbres expansius.

Continua el programa principal per tal que, a partir de `K8`, `(K5_3, K3_5)`, `C8`, `S8`, `W8`, `G0`, `G1`, `G2`, `G3`, escriu la informació anterior en `K8.out`, `K5_3.out`, `C8.out`, `S8.out`, `W8.out`, `graph0.out`, `graph1.out`, `graph2.out`, `graph3.out`.

Compara les distàncies al vèrtex 0 trobades pel mètode de Dijkstra amb els nivell trobats amb els mètodes BFS i DFS en els arbres amb arrel en el vèrtex 0.

Considerem ara els grafs corresponents al moviment de les peces d'un tauler d'escacs $n_1 \times n_2$. Els vèrtexs del graf són les caselles i , per a cada peça, les caselles adjacents a cada casella són aquelles a les quals es pot anar amb un moviment qualsevol de la peça.

Exercici 10 Completa el mòdul `graph.cpp` amb funcions que construeixen els grafs corresponents al moviment de les peces següents: rei (*king*) Kg_{n_1, n_2} , torre (*rook*) Rk_{n_1, n_2} , alfil (*bishop*) Bp_{n_1, n_2} i cavall (*knight*) Kt_{n_1, n_2} , i amb funcions que trobin les distàncies d'una casella a totes les altres.

Continua el programa principal fent un tractament similar del grafs $Kg_{6,6}$, $Rk_{6,6}$, $Bp_{6,6}$ i $Kt_{6,6}$ que l'exercici anterior i trobant el nombre mínim de salts que calen per arribar des de la casella (1, 4) a totes les altres i escrivint els resultats en `Kg6_6.out`, `Rk6_6.out`, `Bp6_6.out` i `Kt6_6.out`.

Les caselles s'haurien d'escriure en coordenades (i_1, i_2) i les informacions relatives als graus de les caselles i de distàncies a una casella donada dels vèrtexs en matrius $n_1 \times n_2$.

A tall d'exemple, les matrius de graus i de distàncies des de la casella (1, 4) del graf associat als moviments del rei en un tauler 6x6 s'escriurien així:

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 3 | 5 | 5 | 5 | 5 | 3 | 4 | 3 | 2 | 1 | 1 | 1 |
| 5 | 8 | 8 | 8 | 8 | 5 | 4 | 3 | 2 | 1 | 0 | 1 |
| 5 | 8 | 8 | 8 | 8 | 5 | 4 | 3 | 2 | 1 | 1 | 1 |
| 5 | 8 | 8 | 8 | 8 | 5 | 4 | 3 | 2 | 2 | 2 | 2 |
| 5 | 8 | 8 | 8 | 8 | 5 | 4 | 3 | 3 | 3 | 3 | 3 |
| 3 | 5 | 5 | 5 | 5 | 3 | 4 | 4 | 4 | 4 | 4 | 4 |

Exercici 11 Estén el programa amb un altre mòdul `wgraphMP.cpp` que contingui funcions que implementin el mètode de Dijkstra de cerca de camins minimal (Minimal Paths) en grafs ponderats i enviïn els resultats a un *stream* de sortida.

Més concretament, es demanen funcions que retornin:

- la distància entre dos vèrtexs, entesa ara com el pes mínim dels camins que els uneixen atenent als pesos de les arestes,
- la distància màxima entre un vèrtex i tots els altres,
- el diàmetre: la distància màxima entre qualsevol parella de vèrtexs,

i funcions que escriguin en l'*stream*:

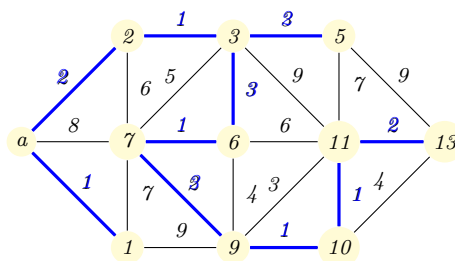
- les distàncies d'un vèrtex donat a tots els altres,
- l'arbre de camins minimal trobat des d'un vèrtex donat i els camins des de tots els vèrtexs accessibles a ell.

Continua el programa principal per tal que, a partir de WK8, (WK5_3, WK3_5), WC8, WS8, WW8, WG0, WG1, WG2, WG3, escrigui en WK8.out, WK5_3.out, WC8.out, WS8.out, WW8.out, wgraph0.out, wgraph1.out, wgraph2.out, wgraph3.out, la informació següent:

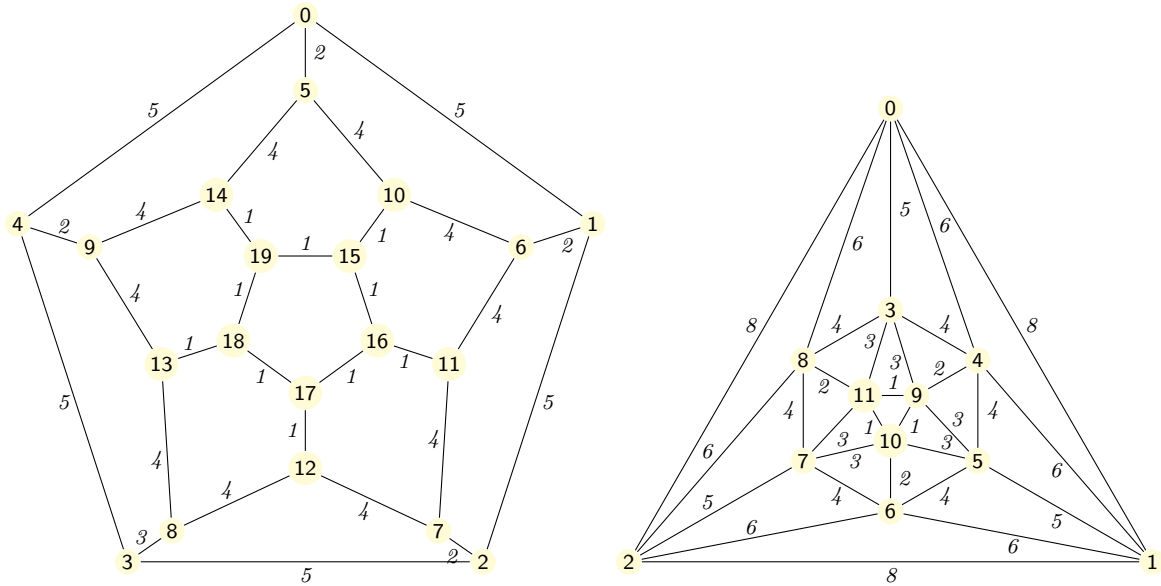
- la distància entre el primer vèrtex i el darrer,
- la distància màxima entre el primer vèrtex i tots els altres,
- les distàncies del primer vèrtex a tots els vèrtexs accessibles, indicant els que ho són,
- el diàmetre,
- els camins de tots els vèrtexs accessibles al primer vèrtex.

Considerem ara el graf ponderat de l'Exemple 3.2.2 dels apunts de teoria.

Exercici 12 Continua el programa comprovant els camins minimal des del vèrtex *a*, assenyalats en blau, i les distàncies a *a* que consten en cada vèrtex:



Exercici 13 Completa el programa, trobant camins minimal a partir del vèrtex 0 a tots els altres vèrtexs en el grafs ponderats següents en el dodecàedre i en l'icosàedre:



Comprimeix els fitxers del programa que has completat (*.h, *.cpp) amb els de dades (*.in) i amb els de solució/projecte (*.sln, *.vcxproj, *.vcxproj.filters), en un fitxer GPr2_CognomsNom.zip, on consti el teu nom en Nom i els teus cognoms en Cognoms, i penja'l en el Campus Virtual en la tasca corresponent a la Pràctica 2. Els fitxers (*.out) i les altres subcarpetes no s'hi haurien d'incloure.