

PRÀCTICA 3: Arbres minimals

En aquesta pràctica cal continuar el projecte anterior `GPr2` en un altre projecte `GPr23`.

L'objectiu és trobar arbres minimals en totes les components d'un graf.

El *mètode de Kruskal* troba les arestes que conformen els arbres minimals a partir d'una aresta minimal i, a cada pas, troba una aresta minimal entre les que no produeixen cicles. Quan no existeix, s'atura el procés. Les arestes considerades poden ser de components diferents del graf i acaben formant arbres minimals en cada component.

El *mètode de Prim* troba les arestes dels arbres minimals, component a component. Parteix d'una aresta minimal inicial que no estigui en cap de les components considerades per tal de construir un arbre minimal de la component a què pertany: a cada pas, parteix d'un arbre en formació i li afegeix, com a branca nova, una aresta minimal que surti de l'arbre i no formi cicle. Quan aquesta aresta no existeix, les arestes afegides conformen un arbre minimal de la component.

Exercici 14 Estén el programa amb un altre mòdul `wgraphMT.cpp` que contingui funcions que implementin els mètodes de Kruskal i de Prim de cerca d'arbres minimals (Minimal Trees) en grafs ponderats. Aquestes funcions haurien de retornar el pes total dels arbres minimals trobats i enviar la informació següent a un *stream* de sortida:

- l'identificador de cada vèrtex, el nombre de component en què es troba i, si escau, l'identificador del seu predecessor en l'arbre minimal;
- els identificadors dels vèrtexs extrems i el pes de les arestes dels arbres minimals; el pes de cada arbre minimal i el pes de tots els arbres minimals.

Continua el programa principal per tal que escrigui aquesta informació dels arbres minimal trobats en els grafs `WK6`, `WC6`, `WS6`, `WW6`, (`WK4_2`, `WK4_2`), `WG0`, `WG1`, `WG2`, `WG2` en els fitxers `WK6.out`, `WC6.out`, `WS6.out`, `WW6.out`, `WK4_3.out`, `wgraph0.out`, `wgraph1.out`, `wgraph2.out`, `wgraph3.out`, respectivament.

Exercici 15 Considera la taula següent de distàncies (en centerars de milles) entre Londres (L), Mèxic DF (Mx), Nova York (NY), París (Pa), Pequín (Pe) i Tòquio (T):

	<i>Mx</i>	<i>NY</i>	<i>Pa</i>	<i>Pe</i>	<i>T</i>
<i>L</i>	56	35	2	51	60
<i>Mx</i>		21	56	68	60
<i>NY</i>			36	68	68
<i>Pa</i>				51	61
<i>Pe</i>					13

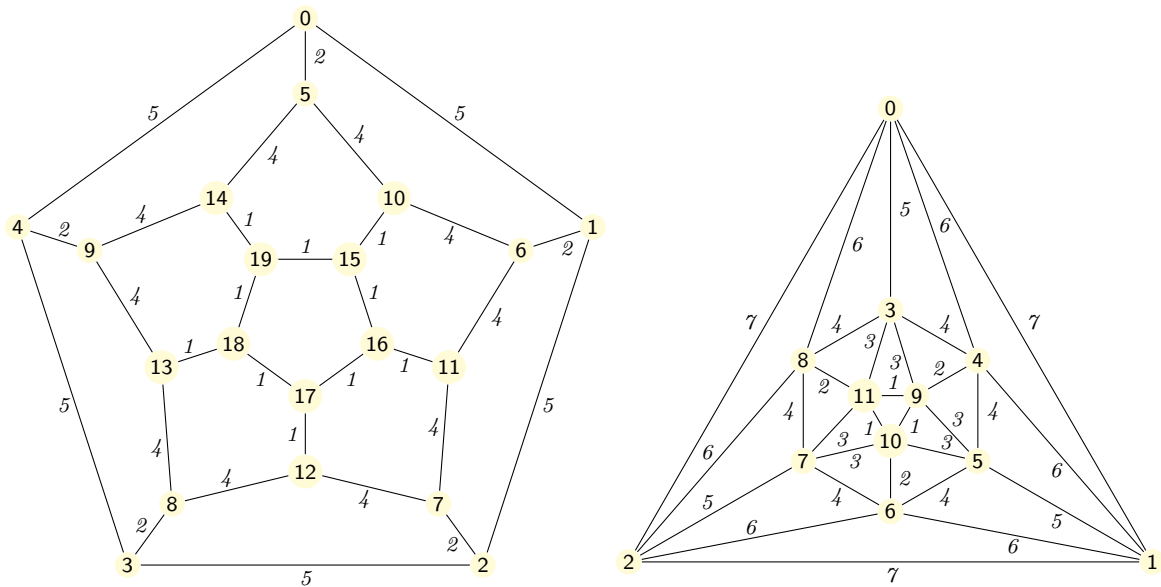
Usa el programa anterior per tal que trobi un arbre minimal de connexió entre totes aquestes ciutats i n'escrigui la informació a `cities1.out`.

Exercici 16 Considera la taula següent de distàncies (en kilòmetres) entre les ciutats xineses de Beijing (B), Chongqing (C), Guangdong (G), Nanjing (N), Shanghai (S), Tianjin (T) i Wuhan (W), i l'estació de generació hidroelèctrica de les Tres Gorges situada a Yichang (Y):

	C	G	N	S	T	W	Y
B	1456	1892	901	1068	111	1056	1116
C		968	1199	1430	1442	650	463
G			1133	1196	1820	836	866
N				266	800	459	626
S					960	681	962
T						988	1080
W							285

Usa el programa anterior per tal que trobi la forma de connexió elèctrica de mínima distància entre totes aquestes ciutats i n'escrigui la informació a `cities2.out`.

Exercici 17 Completa el programa escrivint la informació corresponent al dodecàedre i a l'icosàedre ponderats de la pràctica anterior en `WD.out` i `WI.out`, respectivament:



Comprimeix els fitxers del programa que has completat (*.h, *.cpp) amb els de dades (*.in) i amb els de solució/projete (*.sln, *.vcxproj, *.vcxproj.filters), en un fitxer `GPrac23_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 les Pràctiques 2 i 3. Els fitxers (*.out) i les altres subcarpetes no s'hi haurien d'incloure.