



Documento anónimo

20172.pdf

Enero 2017 - Enunciados + Soluciones PDF



1º Fundamentos de Programación



Grado en Ingeniería Informática



Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación
Universidad de Granada

Como aún estás en la portada, es momento de redes sociales. Cotilléanos y luego a estudiar.



Wuolah



Wuolah



Wuolah_apuntes

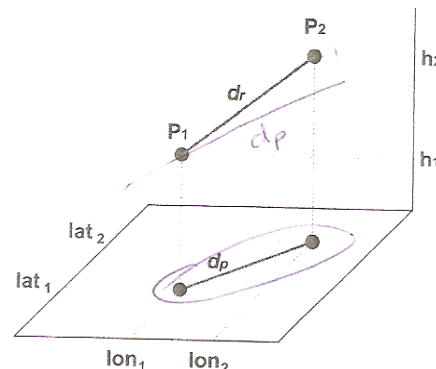
WUOLAH



1. Calcular la distancia *sobre plano* entre dos puntos (distancia que no considera la altura de los puntos). Se trata de la longitud del segmento d_p en la figura de la derecha. Use la llamada *fórmula del Haversine*:

1. Calcular $\Delta_{lon} = lon_2 - lon_1$ y $\Delta_{lat} = lat_2 - lat_1$
2. Calcular $a = \sin^2(\frac{1}{2} \Delta_{lat}) + \cos(lat_1) \cos(lat_2) \sin^2(\frac{1}{2} \Delta_{lon})$
3. Calcular $c = 2 \arcsin(\min(1, \sqrt{a}))$
4. Finalmente, la distancia será $d_p = Rc$

donde $R = 6372797.560856$ metros es la longitud media del radio terrestre. Observe que en las anteriores fórmulas, debe usar los valores de lon y lat expresados en radianes (para ello basta multiplicar los valores expresados en grados por π y dividir por 180).

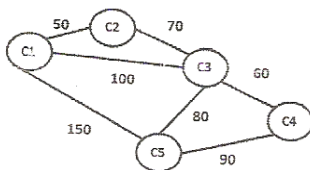


2. Calcular la distancia *real* entre dos puntos (que considera sus alturas). Se trata de la longitud del segmento d_r en la figura de la derecha. Observe que puede formar un triángulo rectángulo a partir de los segmentos d_p , d_r y la diferencia de las alturas.

◁ Ejercicio 4 ▷ clase MapaDistancia

[3.0 puntos]

Se desea construir una clase `MapaDistancia` para almacenar las distancias de los caminos directos que conectan un conjunto de ciudades. Si entre dos ciudades no existe un camino directo, se almacenará un cero. Se supone que la distancia de una ciudad consigo misma será cero y que las distancias son simétricas. Un ejemplo con 5 ciudades sería:



| | C1 | C2 | C3 | C4 | C5 |
|----|-----|----|-----|----|-----|
| C1 | 0 | 50 | 100 | 0 | 150 |
| C2 | 50 | 0 | 70 | 0 | 0 |
| C3 | 100 | 70 | 0 | 60 | 80 |
| C4 | 0 | 0 | 60 | 0 | 90 |
| C5 | 150 | 0 | 80 | 90 | 0 |

Escriba en C++ la clase `MapaDistancia`, indicando los datos miembro e incluyendo solamente las cabeceras de los métodos básicos (Set, Get). Suponga que nunca se trabajará con más de 50 ciudades. A continuación, resuelva las siguientes tareas:

1. Implemente un método que permita obtener la ciudad (su índice) con mayor número de conexiones directas. En el ejemplo, sería la ciudad 3 con 4 conexiones.
2. Implemente un método que, dada una ciudad j , devuelva un objeto de tipo `SecuenciaEnteros` que contenga los índices de las ciudades conectadas directamente con j . Asuma que dispone de la implementación completa de la clase `SecuenciaEnteros`. En el ejemplo, si consideramos la ciudad 4, vemos que está conectada directamente con las ciudades 3 y 5.
3. Implemente un método que, dadas dos ciudades i y j para las cuales no existe un camino directo, devuelva aquella ciudad intermedia z que permita hacer el trayecto entre i y j de la forma más económica posible. Es decir, se trata de encontrar una ciudad z tal que $d(i, z) + d(z, j)$ sea mínima ($d(a, b)$ es la distancia entre las ciudades a y b). El método devuelve -1 si no existe dicha ciudad intermedia. Por ejemplo, si se desea viajar desde la ciudad 2 a la 5, hacerlo a través de la ciudad 1 tiene un costo de $50 + 150 = 200$ mientras que si se hace a través de la ciudad 3, el costo sería $70 + 80 = 150$. Nota: Para la inicialización del mínimo, puede usar la constante `INFINITY`, que garantiza que cualquier dato de tipo `double` es menor que ella.
4. Implemente un método que reciba una `SecuenciaEnteros` conteniendo un conjunto de índices de ciudades, y devuelva `True` si todas las ciudades tienen conexiones directas entre sí. Se devuelve `False` en caso contrario. Muestre un ejemplo de uso.

Si lo considera necesario, puede definir métodos auxiliares. Si lo hace, indique claramente si estos métodos deberían ser públicos o privados.