Calcolo della distanza in chilometri tra due coordinate geografiche con Python

l calcolo della distanza tra due punti sulla superficie terrestre è una problematica comune in molti ambiti, tra cui il GPS, la navigazione e le applicazioni geografiche. Le coordinate geografiche sono rappresentate in termini di latitudine e longitudine, e la sfida consiste nel calcolare la distanza tra questi punti in modo accurato.

In questo articolo, vedremo come utilizzare Python per calcolare la distanza tra due coordinate geografiche, espressa in chilometri. Utilizzeremo diverse metodologie, tra cui la formula dell'Haversine, uno dei metodi più utilizzati per il calcolo delle distanze sferiche.

La formula dell'haversine è un'equazione che consente di calcolare la distanza del percorso più breve (la "distanza ortodromica") tra due punti su una sfera, in particolare sulla superficie terrestre, considerando la curvatura del pianeta.

La formula è la seguente:

$$a = \sin^2(\frac{\Delta\varphi}{2}) + \cos(\varphi_1) \cdot \cos(\varphi_2) \cdot \sin^2(\frac{\Delta\lambda}{2})$$

$$c = 2 \cdot \operatorname{atan2}(\sqrt{a}, \sqrt{1-a})$$

$$d = R \cdot c$$

Per calcolare la distanza tra due coordinate con la formula dell'Haversine, possiamo utilizzare la libreria Python `math` per eseguire i calcoli trigonometrici necessari.

```
import math

def haversine(coord1, coord2):
    # Raggio della Terra in chilometri
    R = 6371.0

    # Coordinate (latitudine e longitudine) in
radianti
    lat1, lon1 = math.radians(coord1[0]),
math.radians(coord1[1])
    lat2, lon2 = math.radians(coord2[0]),
math.radians(coord2[1])

    # Differenze tra le coordinate
    dlat = lat2 - lat1
    dlon = lon2 - lon1

    # Formula dell'Haversine
    a = math.sin(dlat / 2)**2 + math.cos(lat1) *
math.cos(lat2) * math.sin(dlon / 2)**2
    c = 2 * math.atan2(math.sqrt(a), math.sqrt(1 -
a))

    # Distanza finale in chilometri
    distance = R * c
    return distance

# Esempio di utilizzo
coord1 = (41.9028, 12.4964)  # Coordinate di Roma
(latitudine, longitudine)
coord2 = (48.8566, 2.3522)   # Coordinate di Parigi
(latitudine, longitudine)
```

```
distance = haversine(coord1, coord2)
print(f"La distanza tra Roma e Parigi è di circa {distance:.2f} km")
```

Spiegazione del codice:

1. **Importazione della libreria `math`**: questa libreria è necessaria per le funzioni trigonometriche come seno, coseno e arcotangente.
2. **Conversione in radianti**: dato che le funzioni trigonometriche in Python lavorano in radianti, convertiamo le coordinate da gradi a radianti utilizzando la funzione `math.radians()`.
3. **Calcolo della differenza di latitudine e longitudine**: calcoliamo la differenza in latitudine e longitudine tra i due punti.
4. **Applicazione della formula dell'Haversine**: la formula viene applicata per calcolare la distanza in chilometri. La variabile a rappresenta una parte della formula che viene utilizzata nell'arcotangente per ottenere il risultato finale.
5. **Calcolo della distanza**: la distanza finale è ottenuta moltiplicando il raggio della Terra per l'angolo calcolato.

Esistono anche librerie esterne che semplificano il calcolo delle distanze geografiche senza dover implementare manualmente la formula dell'Haversine. Una delle più utilizzate è `geopy`, che fornisce una serie di strumenti per le operazioni geografiche.

Ecco come utilizzare `geopy` per calcolare la distanza tra due coordinate:

```
from geopy.distance import geodesic

# Coordinate di Roma e Parigi
```

```
coord1 = (41.9028, 12.4964)
coord2 = (48.8566, 2.3522)

# Calcolo della distanza in chilometri
distance = geodesic(coord1, coord2).kilometers
print(f"La distanza tra Roma e Parigi è di circa
{distance:.2f} km")
```

geopy utilizza un modello ellissoidale per la Terra, che fornisce una stima più accurata rispetto alla formula dell'Haversine.

Conclusione

Il calcolo della distanza tra due coordinate geografiche è una delle applicazioni più comuni nei sistemi di navigazione e nelle app di mappatura. Python, con l'uso della formula dell'Haversine o librerie come geopy, rende questo calcolo semplice e preciso.