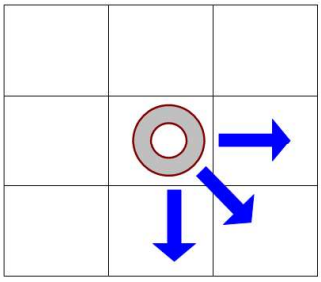
Robot pulisci-pavimenti



Il software di controllo di un robot pulisci-pavimenti registra la posizione del robot ogni secondo, registrando le coordinate x e y del robot misurate rispetto alla sua base di ricarica posizionata nella cella (0,0) nell'angolo in alto a sinistra di una stanza che misura 3m x 3m. Il robot ogni secondo può solo rimanere nella stessa posizione o spostarsi verso il basso e/o verso destra. Questi dati sono registrati in un dizionario `posizioni` in cui ogni elemento ha come chiave il tempo in secondi trascorso dall'accensione e come valore la tupla delle coordinate (x,y).

Iniziamo a crearci i dati di esempio considerando che il robot si sia spostato per 20 secondi.

```
1 durata = 20
2 x_array = np.cumsum(np.random.randint (0, 2, size = durata))
3 y_array = np.cumsum(np.random.randint (0, 2, size = durata))
4 posizioni = {}
5 posizioni[0] = (0,0)    #posizione iniziale
6 for i,(x,y) in enumerate(zip(x_array,y_array)):
7     posizioni[i + 1]= (x,y)
8 print(posizioni)
```

```
{0: (0, 0), 1: (0, 1), 2: (0, 2), 3: (1, 2), 4: (1, 2), 5: (2, 3), 6: (2, 4), 7: (2, 5),
8: (2, 6), 9: (2, 6), 10: (2, 6), 11: (3, 7), 12: (4, 7), 13: (5, 8), 14: (6, 9), 15: (6,
9), 16: (6, 9), 17: (6, 10), 18: (6, 10), 19: (6, 10), 20: (6, 11)}
```

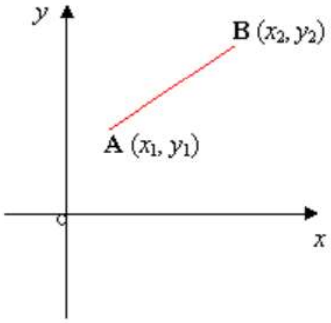
All'inizio ovvero al tempo 0, la posizione del robot è alla base con coordinate (0,0), mentre al tempo 20 è nella sua ultima posizione.

Vediamo in dettaglio il codice alle righe #2 #3

- con `np.random.randint` (0, 2, size = durata) si creano 20 numeri casuali tra 0 (non si muove) e 1 (si sposta)
- con `np.cumsum()` che calcola per ogni elemento la somma dei valori fino a quel indice (per esempio `np.cumsum([1,2,3])` restituisce `[1,3,6]`). In questo modo otteniamo le coordinate di x e y che si raggiungono sommando successivamente i movimenti ottenuti casualmente con la funzione `randint`

Ora dobbiamo:

1. creare una matrice avente 2 colonne e 21 righe (posizioni contiene 21 coppie di coordinate) contenente tutte le coordinate raggiunte dal robot;
2. calcolare quanto spazio ha percorso il robot, sapendo che le coordinate sono espresse in decimetri. Si ipotizza che il robot segua una traiettoria rettilinea tra un punto ed il successivo per cui per calcolare la distanza tra le varie coordinate dobbiamo utilizzare la formula:



(1) FORMULA GENERALE

$$d = \overline{AB} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

oppure

$$d = \overline{AB} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Usando Python e le liste

```python
import math

#1)Creare una matrice contentente tutte le coordinate raggiunte dal robot.
matrice_posizioni = [[x,y] for _,(x,y) in posizioni.items()]
print(matrice_posizioni[:5])

#2)Sapendo che le coordinate sono espresse in decimetri calcolcare quanto spazio ha percorso il robot.
spostamento = 0
for i, el in enumerate (matrice_posizioni[:-1]):
    spostamento += math.sqrt((matrice_posizioni[i+1][0] - el[0])**2 + (matrice_posizioni[i+1][1] - el[1])**2)

print(f"Lo spostamento totale è di {spostamento * 0.1} metri")
```

```
[[0, 0], [0, 1], [1, 2], [2, 2], [2, 3]]
Lo spostamento totale è di 1.4656854249492384 metri
```

usando Numpy

```python
#1)Creare una matrice contentente tutte le coordinate raggiunte dal robot.
matrice_posizioni = np.array(list(posizioni.values()))
print(matrice_posizioni[:5])

#2)Sapendo che le coordinate sono espresse in decimetri calcolcare quanto spazio ha percorso il robot.
matrice_posizioni_traslata = np.roll(matrice_posizioni,-1,axis = 0)
spostamenti = matrice_posizioni_traslata[:-1] - matrice_posizioni[:-1]
print(f"Lo spostamento totale è di {np.sum(np.sqrt(np.sum(spostamenti**2, axis=1)))*0.1} metri")
```

```
[[0 0]
 [0 1]
 [1 2]
 [2 2]
 [2 3]]
Lo spostamento totale è di 1.4656854249492381 metri
```

Analizziamo il codice:

- nella riga #2 si crea la matrice ottenendo la lista dei valori presenti nel dizionario. Se potevamo usare direttamente gli array degli spostamenti, avremmo scritto:

```
matrice_posizioni = np.concatenate(([[0,0]], np.stack((x_array,y_array), axis=1)), axis=0)
```

- nella riga #6 con la funzione `np.roll(..., -1, axis=0)` otteniamo la matrice con le righe traslate di una posizione a sinistra in modo da poter fare il calcolo degli spostamenti nella riga #7 di ogni coordinata rispetto al punto precedenti. Potevamo anche calcolarli direttamente con

```
spostamenti = matrice_posizioni[1:] - matrice_posizioni[:-1]
```

- nella riga #8 con `spostamenti**2` otteniamo l'array con tutte le differenze delle coordinate elevate alla seconda. Quindi con `np.sum(spostamenti**2, axis=1))` otteniamo un array con le somme di ogni coppia di quadrati ottenuti. Con `np.sqrt()` otteniamo l'array con tutte le distanze espresse in decimetri. Non ci resta che sommarle con `np.sum()` e moltiplicare il risultato per `0.1` per ottenere il valore corrispondente in metri.

Mettiamoci alla prova

1. Due giocatori si affrontano in una partita a dadi: ogni volta che un giocatore ottiene un numero maggiore del suo rivale guadagna un punto, mentre se entrambi i giocatori ottengono lo stesso numero guadagnano mezzo punto ciascuno.
    - Simulare una partita di 1000 lanci e registrare i risultati di 10 partite.
    - Se un giocatore avesse un dado truccato in cui il 5 e il 6 hanno il doppio di probabilità di uscire rispetto agli altri numeri, ce ne potremmo accorgere?
2. In riferimento all'esercizio della centralina ambientale montata nel parco della scuola
    - Ripulire i dati dai valori errati e calcolare media, massimo e minimo.
    - Individuare se ci sono due dati errati in finestre di al massimo 5 misurazioni consecutive e in tal caso segnalare che la centralina deve essere mandata in riparazione.
3. Sul territorio comunale sono installate tre centraline per la misurazione del livello di polveri sottili PM10 nell'aria. Talvolta le centraline vanno in uno stato

di errore per cui la misurazione delle PM10 fallisce ed il dato corrispondente riporta il valore 9999.9 all'interno dell'array delle misurazioni. Inoltre, per motivi sconosciuti, talvolta le centraline riportano il valore -1. Statisticamente si è visto che il primo errore ha una frequenza dello 0.2% mentre il secondo del 0.1%

- Creare una matrice avente 3 colonne e le righe contenente i dati di tutte le centraline registrati nelle 24 ore (si registra il dato ogni minuto)
- Eliminare tutte le righe della matrice in cui vi sia almeno una misura errata o una misura di origine sconosciuta