Exercise 5

Table of Contents

1. Lösungsidee

Zum Generieren der Walks wird die monte_carlo_walk Funktion aufgerufen. Es wird berechnet, wie viele Walks eine Distanz 6 aufweisen. Anhand von diesem wird die Wahrscheinlichkeit berechnet, ob ein Walk mit maximaler Distanz kurz ist. Dies wird dann für jede Länge ausgegeben.

2. Fragen

2.1. How do you determine that your results are stable with respect to the number of repetitions that are performed for each maximum length?

Da für jede Länge die Generierung eines Walks 10000 mal wiederholt wird, entstehen sehr stabile Ergebnisse. Schwankungen werden durch die vielen Wiederholen ausgeglichen.

2.2. What happens if you change the condition for a walk being short from having a distance of smaller or equal six to other numbers?

Wenn die Bedingung verändert wird, verschiebt sich einfach die Wahrscheinlichkeit entsprechend. Wenn statt 6 5 als Bedingung verwendet wird, verschieben sich einfach die Wahrscheinlichkeiten um 1. Walks, welche kleiner sind als die Bedingung sind sowieso zu 100% kurz.

2.3. Can you identify a pattern for even and odd numbers?

Walks mit gerader Länge sind mit einer höheren Wahrscheinlichkeit kurz als die der mit ungeraden Längen. Dies liegt daran, dass für den Vergleich eine gerade Zahl verwendet wurde. Wird hier eine ungerade Zahl verwendet, dreht sich dies um und die ungeraden Walks weißen eine höhere Wahrscheinlichkeit auf.

3. Test

Mit welcher Wahrscheinlichkeit weisen Walks eine Länge 6 auf?

```
Walks with length of 1 are 100.00% short
Walks with length of 2 are 100.00% short
Walks with length of 3 are 100.00% short
Walks with length of 4 are 100.00% short
Walks with length of 5 are 100.00% short
Walks with length of 6 are 100.00% short
Walks with length of 7 are 97.09% short
Walks with length of 8 are 98.63% short
Walks with length of 9 are 92.32% short
Walks with length of 10 are 95.64% short
Walks with length of 11 are 87.35% short
Walks with length of 12 are 92.24% short
Walks with length of 13 are 82.16% short
Walks with length of 14 are 88.86% short
Walks with length of 15 are 77.75% short
Walks with length of 16 are 85.37% short
Walks with length of 17 are 73.21% short
Walks with length of 18 are 81.64% short
Walks with length of 19 are 68.80% short
Walks with length of 20 are 78.45% short
Walks with length of 21 are 65.77% short
Walks with length of 22 are 75.77% short
Walks with length of 23 are 61.44% short
Walks with length of 24 are 71.61% short
Walks with length of 25 are 60.00% short
Walks with length of 26 are 69.52% short
Walks with length of 27 are 56.74% short
Walks with length of 28 are 66.91% short
Walks with length of 29 are 54.18% short
Walks with length of 30 are 63.43% short
Walks with length of 31 are 52.36% short
Walks with length of 32 are 61.93% short
Walks with length of 33 are 50.61% short
Walks with length of 34 are 59.42% short
Walks with length of 35 are 47.36% short
Walks with length of 36 are 58.11% short
Walks with length of 37 are 45.53% short
Walks with length of 38 are 55.70% short
Walks with length of 39 are 44.37% short
Walks with length of 40 are 53.57% short
Walks with length of 41 are 42.07% short
Walks with length of 42 are 52.03% short
Walks with length of 43 are 40.71% short
Walks with length of 44 are 49.94% short
Walks with length of 45 are 40.26% short
Walks with length of 46 are 49.49% short
Walks with length of 47 are 38.38% short
Walks with length of 48 are 47.16% short
Walks with length of 49 are 36.97% short
Walks with length of 50 are 46.64% short
```

4. Quellcode

```
import Memory as mem

if __name__ == "__main__":
    max_walk_length = 6
    walks = mem.monte_carlo_walk(50)
    for length, walks in walks.items():
        walks_with_smaller_distance = 0
        for _, distance, in walks:
                if distance <= max_walk_length:
                    walks_with_smaller_distance += 1
        print(f"Walks with length of {length} are {(walks_with_smaller_distance / len(walks)) *
100:.2f}% short")
```