

Queueing Theory HW2

110703007 資科三 鄭睿宏

```
stateDiagram
    state fork_state <<fork>>
    fork_state

    A: Rope < 50cm
    note left of fork_state
        Short rope
    end note

    note right of fork_state
        Long rope
    end note

    B: Rope > 50cm
```

題目說明



假設有一個繩子長度是 100 公分，隨機(uniformly distributed)從任意地方剪斷成兩段，我們取這兩段中比較短的一段。請問比較短的這段的期望值(expected value)會是幾公分長？

算法說明

- 若是一根繩子100公分，任意截斷繩子，繩子必分為長度為 x 和 $100-x$ 的兩段繩
- 由於短繩比長繩短，令短繩為 x ，可得 $x \leq 100 - x \rightarrow 2x \leq 100 \rightarrow x \leq 50$
- 短繩最短 0 cm，最長50 cm

- expectation = $\sum P(x = s)s \, dx$

Execution Video:

<https://youtu.be/EB-JQAfokIM>

Execution Result:

```
PS E:\project\Queueing-Theory\hw2> python .\hw2.py
pick the short rope for 1000 times
Length expectation of short : 24.3794
• PS E:\project\Queueing-Theory\hw2> python .\hw2.py
pick the short rope for 100000 times
Length expectation of short : 25.0181
• PS E:\project\Queueing-Theory\hw2> python .\hw2.py
pick the short rope for 100000 times
• Length expectation of short : 24.9741
```

Code

```
import numpy.random as rd

def pick_rope(pickShort=True, num_iterations=1000):
    expectation_sum = 0

    for iter in range(num_iterations):
        # random cut the rope from 0 to 100
        rope = rd.uniform(0, 100)
        # pick short rope: rope that shorter than 50
        shortRope = rope if rope <= 50 else 100 - rope
        # summary all the short rope
        expectation_sum += shortRope
        print(f"pick the short rope for {iter+1} times", end
              ="\r")
```

```
    print(f"\nLength expectation of short : {expectation_sum/  
num_iterations:.4f} ")  
  
if __name__ == '__main__':  
    pick_rope(num_iterations=100000)
```