

Accordo Bizantino MC

Lorenzo Livio Vaccarecci (matr. 5462843)

A.A. 2023/2024

1 Codice

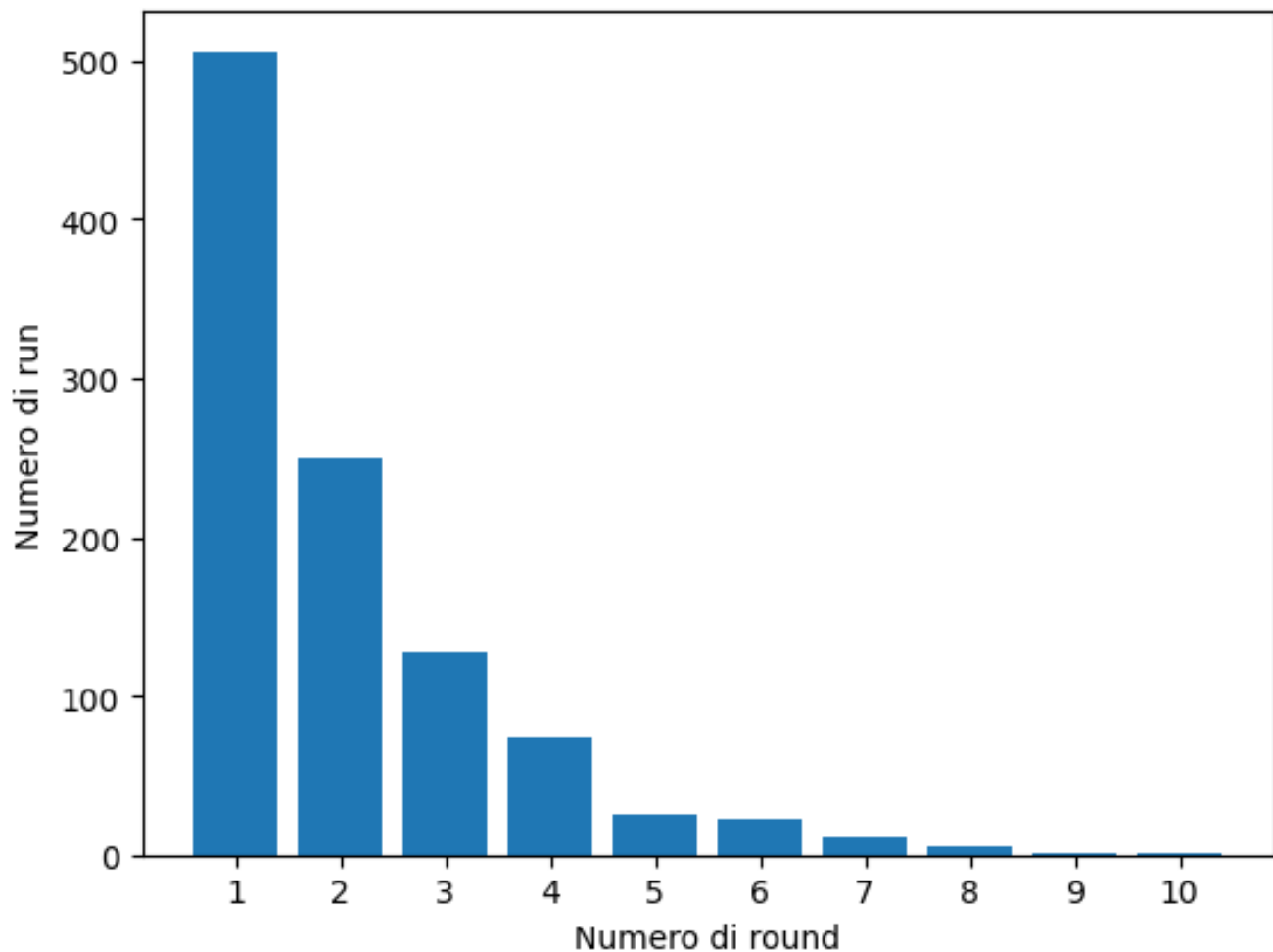
```
1 import random
2 import matplotlib.pyplot as plt
3
4 def tally(j, maj, matrix):
5     eq = 0
6     for i in range(0, n):
7         if matrix[j][i] == maj:
8             eq += 1
9     return eq
10
11 def maj(j, matrix):
12     max = [0,0]
13     for i in range(0, n):
14         if matrix[j][i] == 0:
15             max[0] += 1
16         else:
17             max[1] += 1
18     return 0 if max[0] > max[1] else 1
19
20 def coin():
21     return random.randint(0, 1)
22
23 def MCByzantineGeneral(matrix):
24     matrix_copy = [row[:] for row in matrix]
25     for r in range(10):
26         coinFlip = coin()
27
28         for j in range(n):
29             for i in range(n):
30                 if i != j:
31                     matrix_copy[i][j] = matrix_copy[j][j]
32
33         for j in range(n):
34             for i in range(n):
35                 if i != j:
36                     matrix_copy[j][i] = matrix_copy[i][j]
37
38
39
40
```

```

41     success = False
42     for j in range(n):
43         for i in range(n):
44             majVar = maj(j, matrix_copy)
45             if tally(j, majVar, matrix_copy) >= T:
46                 matrix_copy[i][j] = majVar
47                 if majVar == coinFlip:
48                     success = True
49                     break
50             else:
51                 matrix_copy[i][j] = coinFlip
52
53     if success:
54         roundSucc[r] += 1
55     return
56
57 R=2**10
58 n=4
59 f=1
60 T=(2*f)+1
61
62 bitSndRcv = [[0 for _ in range(n)] for _ in range(n)]
63 for i in range(0, n):
64     for j in range(0, n):
65         if i == 3:
66             bit = 1 - bitSndRcv[j][i]
67         else:
68             bit = random.randint(0, 1)
69         bitSndRcv[i][j]=bit
70
71 roundSucc = [0 for _ in range(10)]
72 for i in range(R): # Run
73     MCByzantineGeneral(bitSndRcv)
74
75 plt.bar(list(range(1,len(roundSucc)+1)), roundSucc)
76 plt.xlabel("Numero di round")
77 plt.ylabel("Numero di run")
78 plt.show()

```

2 Osservazioni



Il grafico suggerisce che la maggior parte delle 2^{10} esecuzioni dell'accordo bizantino viene completata nel primo round, con un dimezzamento del successo ad ogni round successivo.

Se in un qualche round per un processo affidabile j risulta che $tally(j) \geq T$ e l'esito del lancio della moneta coincide con $maj(j)$, il round successivo l'accordo è certamente raggiunto perchè significa che almeno T processi hanno raggiunto l'accordo su un valore e , se coincide con il lancio della moneta, nel round successivo il valore verrà propagato ai processi rimanenti facendo raggiungere l'accordo.