

# Relazione: LS Quicksort

Lorenzo Livio Vaccarecci (matr. 5462843)

A.A. 2023/2024

```
1 import random
2 from tqdm import tqdm
3 import matplotlib.pyplot as plt
4 import math
5
6 X = 0
7
8 def confronta(s, pos, Sequence):
9     global X
10    sequenceMin = []
11    sequenceMax = []
12    for i in range(len(Sequence)):
13        if (i != pos):
14            if Sequence[i] < s:
15                sequenceMin.append(Sequence[i])
16                X += 1
17            else:
18                sequenceMax.append(Sequence[i])
19                X += 1
20    return [sequenceMin, sequenceMax]
21
22 def LVQuickSort(Sequence):
23     if (len(Sequence) <= 1):
24         return Sequence
25     pos = random.randint(0, len(Sequence)-1)
26     s = Sequence[pos]
27     confronto = confronta(s, pos, Sequence)
28     sequenceMin = LVQuickSort(confronto[0])
29     sequenceMax = LVQuickSort(confronto[1])
30     sorted_sequence = sequenceMin
31     sorted_sequence.extend([s])
32     sorted_sequence.extend(sequenceMax)
33     return sorted_sequence
34
35 def random_array(n):
36     arr = []
37     for i in range(n):
38         arr.append(random.randint(0, n))
39     return arr
40
41 def valore_medio(R, Xr):
42     sommatoria = sum(Xr)
43     return (1/R) * sommatoria
```

```

44
45 def deviazione_standard(R, Xr, u):
46     sommatoria = 0
47     for i in Xr:
48         sommatoria += ((i-u)**2)
49     return (1/(R-1)) * sommatoria
50
51 def markov(mu, val):
52     return mu/(val*mu)
53
54 def chebyshev(mu, val, dev):
55     return dev / (((val-1)**2)*(mu**2))
56
57 n = 10**4
58 R = 10**5
59 Xr = []
60
61 for i in tqdm(range(R)):
62     array = random_array(n)
63     X = 0
64     LVQuickSort(array)
65     Xr.append(X)
66
67 val_medio = valore_medio(R, Xr)
68 dev_standard = deviazione_standard(R, Xr, val_medio)
69
70 print("Valore medio: " + str(val_medio))
71 print("Varianza: " + str(dev_standard))
72 print("Deviazione standard: " + str(math.sqrt(dev_standard)))
73
74 plt.hist(Xr, edgecolor="black", bins=50)
75 plt.xlabel("Numero di confronti")
76 plt.ylabel("Frequenza")
77 plt.show()
78
79 v1 = 2
80 v2 = 3
81
82 print(markov(val_medio, v1))
83 print(markov(val_medio, v2))
84
85 print(chebyshev(val_medio, v1, dev_standard))
86 print(chebyshev(val_medio, v2, dev_standard))

```

Dopo aver implementato l'algoritmo Quicksort Las Vegas, si sono utilizzate delle sequenze di numeri casuali di cardinalità  $10^4$  eseguendo l'algoritmo per  $R = 10^5$  volte (con array generati in modo casuale) e si sono ottenuti i seguenti valori:

- Valore medio( $\hat{\mu}$ )  $\simeq 156533.23$
- Deviazione standard empirica( $\hat{\sigma}$ )  $\simeq 6498.49$

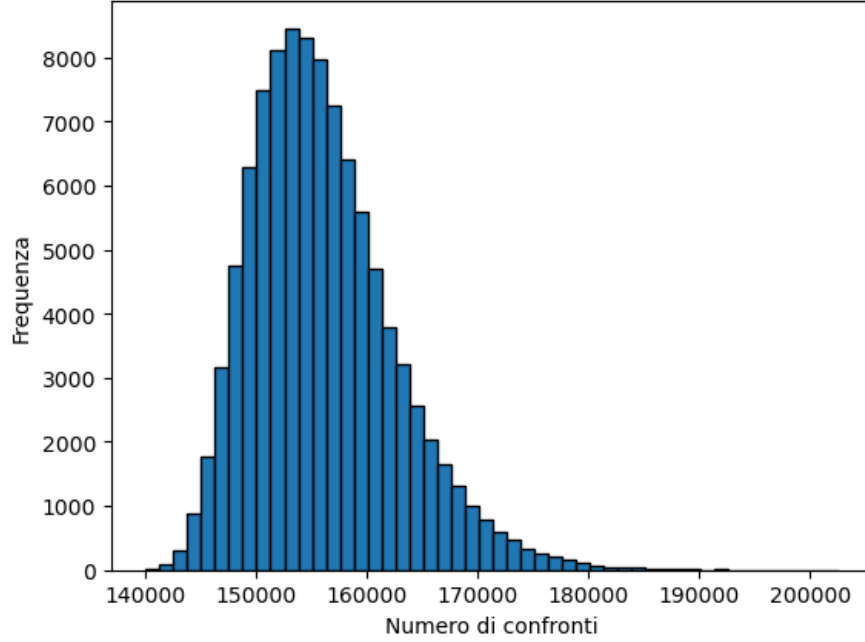
Per calcolare i valori sono state usate le formule:

$$\hat{\mu} = \frac{1}{R} \sum_{r=1}^R X_r$$

Dove  $X_r$  è il numero di confronti effettuati al passo  $r$ -esimo e:

$$\hat{\sigma}^2 = \frac{1}{R-1} \sum_{r=1}^R (X_r - \hat{\mu})^2$$

Usando il numero di confronti effettuati si può generare il seguente istogramma:



Il grafico mostra che la distribuzione dei valori è ampia (questo conferma la deviazione standard elevata), il picco si trova in un valore vicino a 150000 confermando il valore medio ottenuto. Un'ulteriore conferma che il valore medio è quello atteso è data dalla formula:

$$\mathbb{E}[X] = \sum_{i=1}^{n-1} 2 \ln(n - i + 1) = 2 \sum_{i=1}^{n-1} \ln(n - i + 1) = 2 \sum_{i=1}^{10^4-1} \ln(10^4 - i + 1) \simeq 164216.47$$

In modo simile:

$$\mathbb{E}[X] = 2n \ln(n) = 2 \cdot 10^4 \ln(10^4) \simeq 184206.81$$

Si può notare che i valori medi ottenuti dalle formule teoriche e quello calcolato dall'implementazione dell'algoritmo sono molto simili. Questa discrepanza può essere attribuita a diversi fattori, tra cui la natura approssimativa delle formule teoriche e la presenza di numeri casuali. Tuttavia, il valore medio ottenuto dal programma è ragionevole e conferma l'efficienza dell'algoritmo.

Calcolata la disuguaglianza di Chebyshev usando la formula:

$$\frac{\hat{\sigma}^2}{(v-1)^2 \hat{\mu}^2}$$

Abbiamo che per il doppio ( $2\hat{\mu} \simeq 313066.46$ ) e il triplo ( $3\hat{\mu} \simeq 469599.69$ ) del valore medio con  $v = 2$   $v = 3$  rispettivamente:

$$\frac{6498.49^2}{(2-1)^2 \cdot 156533.23^2} \simeq 0.0017 \simeq 0.17\%$$

e

$$\frac{6498.49^2}{(3-1)^2 \cdot 156533.23^2} \simeq 0.0004 \simeq 0.04\%$$

Calcolata anche la disuguaglianza di Markov per le stesse  $v$  usando la formula:

$$\frac{\hat{\mu}}{v\hat{\mu}}$$

Si ottengono facilmente le due probabilità:

$$\frac{156533.23}{2 \cdot 156533.23} = 0.5 = 50\%$$

e

$$\frac{156533.23}{3 \cdot 156533.23} \simeq 0.33 \simeq 33\%$$

Secondo la disuguaglianza di Markov, la probabilità che il numero di confronti sia superiore a  $2\hat{\mu}$  è del 50%, mentre la probabilità che sia superiore a  $3\hat{\mu}$  è del 33%. Questo significa che la disuguaglianza di Markov è molto meno cauta rispetto a quella di Chebyshev che, invece, fornisce un limite superiore alla probabilità che il numero di confronti sia superiore a  $2\hat{\mu}$  del 0.17% e del 0.04% per  $3\hat{\mu}$ .

In conclusione, i risultati ottenuti dall'implementazione dell'algoritmo Quicksort Las Vegas confermano la sua efficienza per la risoluzione di problemi di ordinamenti di array molto grandi. Il valore medio e la deviazione standard calcolati indicano che la distribuzione dei valori è ampia ma concentrata attorno al valore medio come confermato dall'istogramma.

Le disuguaglianze di Chebyshev e Markov confermano che la probabilità che il numero di confronti sia superiore al doppio e al triplo del valore medio è molto bassa.