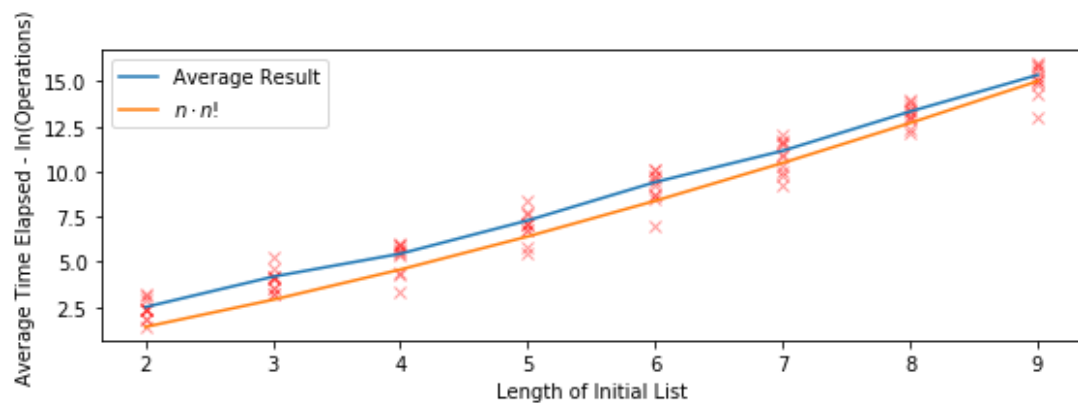


一、bogo-sort 介紹

worst case performance of $O(\infty)$, a best case performance of $O(n)$, and an average performance of $O(n \cdot n!)$. 已有許多現成的程式。

二、實驗



三、分析時間

Best case:一開始就是 sorted 的狀態所以為 $O(n)$

Worst case:永遠隨機不到 sorted 的狀態所以為 $O(\infty)$

Average performance: 期望的位置交換次數漸近 $(n-1)n!$ 所以為 $O(n \cdot n!)$

四、我設計了一個 Boooooogo sort

Pseudo-code:

```
while not isSorted(deck) or not isSameElement(deck):
```

```
    for i in range length of deck
```

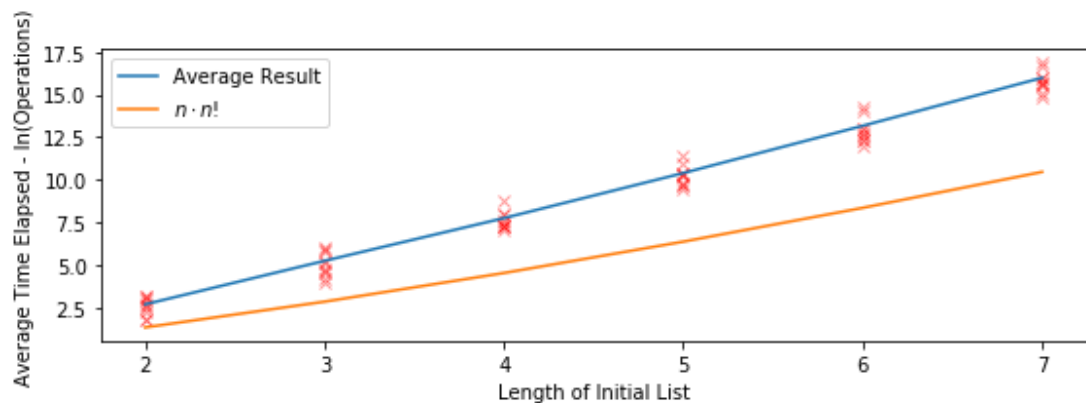
```
        newDeck[i] = chose a random element in deck
```

```
    deck = newDeck
```

由於我設計的演算法在 random 一個新的排列時有可能會選到重複的
元素，所以 Average performance 會是 $O(n \cdot n^n)$ ，Best、Worst 和 bogo-
sort 一樣。

五、 附件 boooooogosort.py

六、 實驗



七、 由於 boooooogo-sort 時間使用上升太快，在 list 長度 9 得時候跑了 3 小
時，然而圖沒輸出就 crash 了，所以只有長度到 7 的圖，時間增加可能不
太明顯。

手機:0975968676

Email:fisurpco3@gmail.com

資料來源:

<https://en.wikipedia.org/wiki/Bogosort>

<https://github.com/TheAlgorithms/Python/blob/master/sorts/bogosort.p>

y