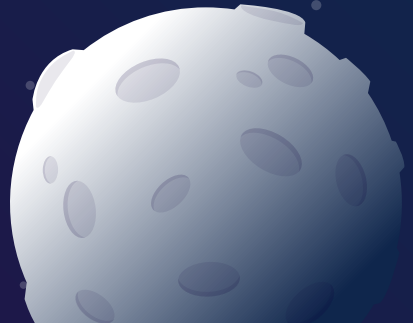
Una Comparación entre C++, Go y Python



Introducción

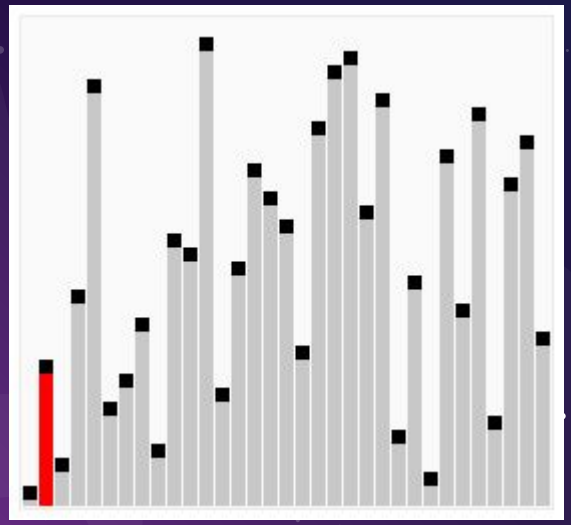
Go es un lenguaje compilado basado en C y con la facilidad de Python.

C++ es un lenguaje compilado, extensión de C.

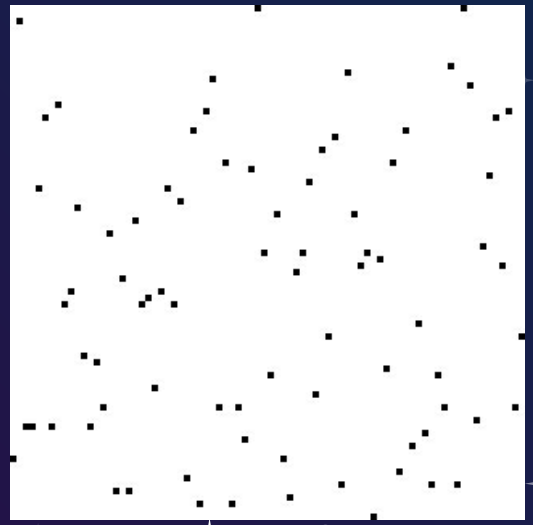
Python es un lenguaje interpretado, con uso científico.

Se pondrá a prueba los 3 lenguajes mencionados con los algoritmos de ordenamiento:

Cocktail Sort



Counting Sort



ALGORITMOS

Cocktail Sort

1. El primer paso es recorrer la lista de datos de izquierda a derecha, se comparan los valores adyacentes, y si el valor de la izquierda es mayor, se intercambian los valores. El objetivo es poner el valor máximo al final.

2. El segundo paso es recorrer la lista de datos de derecha a izquierda, se comparan los valores adyacentes, y si el valor de la derecha es menor, se intercambian los valores. El objetivo es poner el valor mínimo al inicio.

Counting Sort

1. Busca el elemento máximo (max) y el mínimo (min).
2. Crea una lista auxiliar de tamaño max + 1.
3. Se subdivide en los siguientes pasos:

   - Se almacena el recuento de cada elemento de la lista en su índice correspondiente en la lista auxiliar.

   - Se suman los recuentos (a[i] + a[i-1]]), esto para colocar los elementos en el índice correcto de la lista ordenada.

4. Posiciona los elementos en la lista original o una de salida, para esto se utilizan los índices.

Cocktail Sort

Caso	Complejidad
Mejor caso	$O(n)$
Caso promedio	$O(n\ 2\ )$
Peor caso	$O(n\ 2\ )$.

Counting Sort

Caso	Complejidad
Mejor caso	$O(n$+k).
Caso promedio	$O(n$+k)
Peor caso	$O(n$+k)

```cpp
vector<int> cocktailSort(vector<int> list) {
  int last = list.size() - 1;
  while(true) {
    bool swapped = false;
    for (int i = 0; i < last; i++) {
      if (list[i] > list[i+1]) {
        int a = list[i];
        list[i] = list[i+1];
        list[i+1] = a;
        swapped = true;
      }
    }
    if(!swapped) return list;

    swapped = false;
    for (int i = last - 1; i >= 0; i--) {
      if (list[i] > list[i+1]) {
        int a = list[i];
        list[i] = list[i+1];
        list[i+1] = a;
        swapped = true;
      }
    }
    if (!swapped) return list;
  }
  return list;
}
```

```cpp
vector<int> CountingSort(vector<int> list) {
  int minItem = *min_element(list.begin(), list.end());
  vector<int> countList ((*max_element(list.begin(), list.end()) - minItem + 1), 0);
  vector<int> outputList (list.size(),0);

  for(int i = 0; i < list.size(); i++) countList[list[i] - minItem] += 1;

  for(int i = 1; i < countList.size(); i++) countList[i] += countList[i-1];

  for (int i = list.size() - 1; i >= 0; i--) {
    outputList[countList[list[i] - minItem] - 1] = list[i];
    countList[list[i] - minItem] -= 1;
  }

  for(int i = 0; i < list.size(); i++) list[i] = outputList[i];

  return list;
}
```

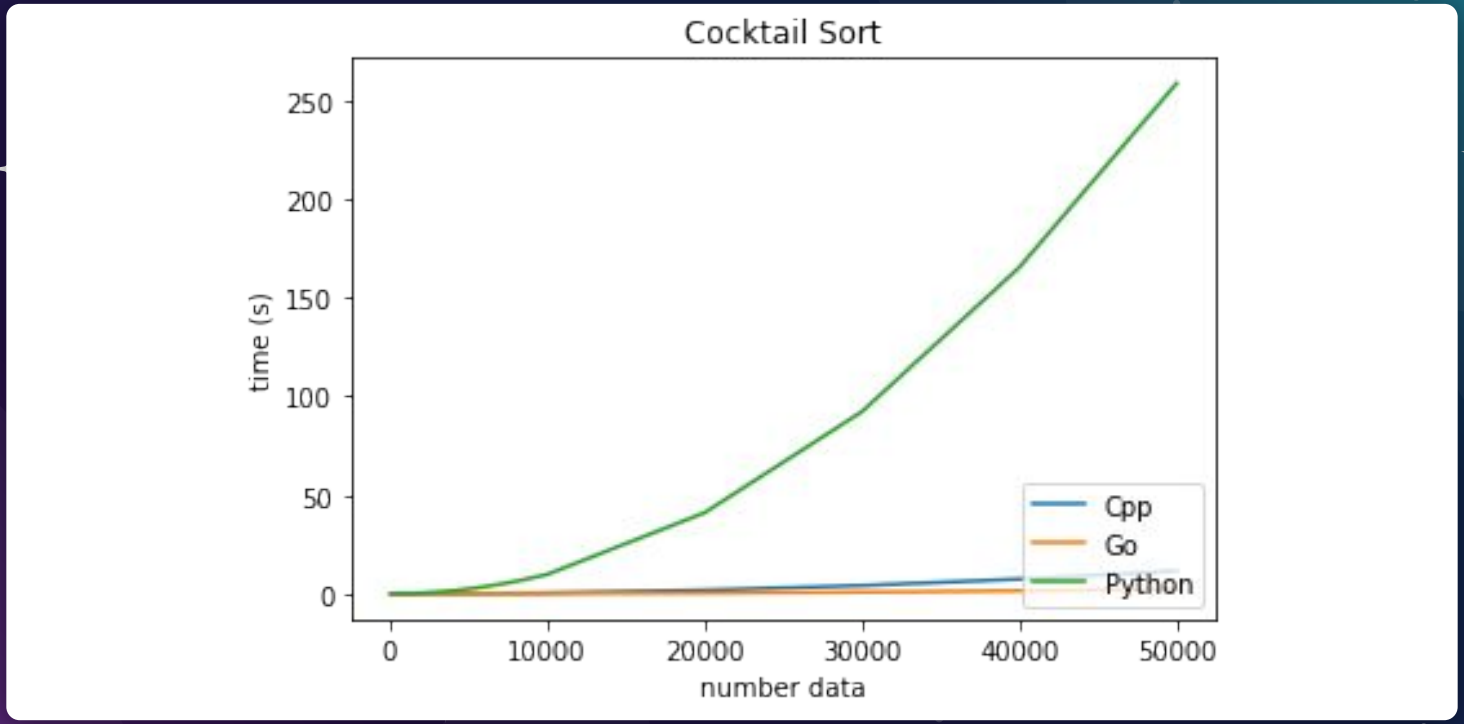
```go
func cocktailSort(list []int) {
    last := len(list) - 1
    for {
        swapped := false
        for i := 0; i < last; i++ {
            if list[i] > list[i+1] {
                list[i], list[i+1] = list[i+1], list[i]
                swapped = true
            }
        }
        if !swapped {
            return
        }
        swapped = false
        for i := last - 1; i >= 0; i-- {
            if list[i] > list[i+1] {
                list[i], list[i+1] = list[i+1], list[i]
                swapped = true
            }
        }
        if !swapped {
            return
        }
    }
}
```

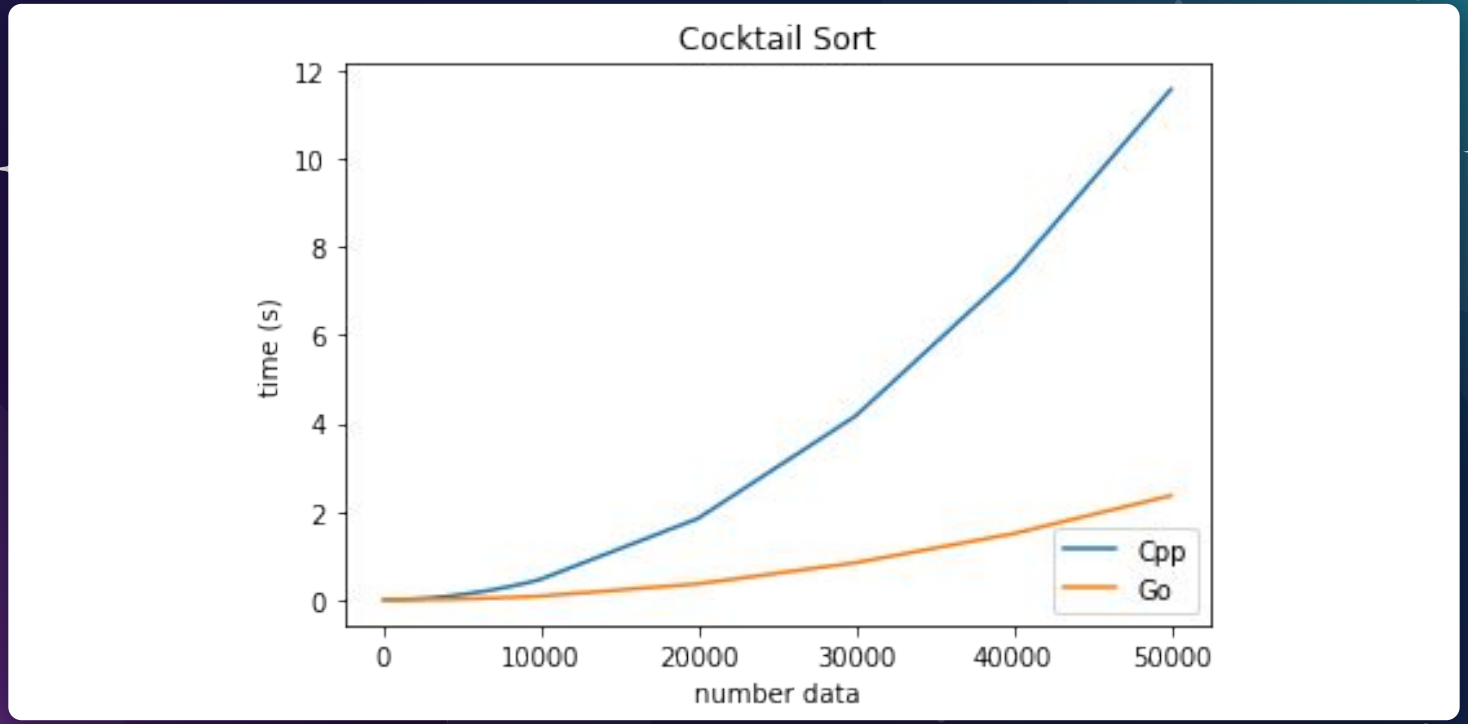
```go
func countingSort(list []int) {
    minItem := Min(list)
    countList := Zero(Max(list) - minItem + 1)
    outputList := Zero(len(list))

    for i := 0; i < len(list); i++ {
        countList[list[i]-minItem] += 1
    }

    for i := 1; i < len(countList); i++ {
        countList[i] += countList[i-1]
    }

    for i := len(list) - 1; i >= 0; i-- {
        outputList[countList[list[i]-minItem]-1] = list[i]
        countList[list[i]-minItem] -= 1
    }

    for i := 0; i < len(list); i++ {
        list[i] = outputList[i]
    }
}
```

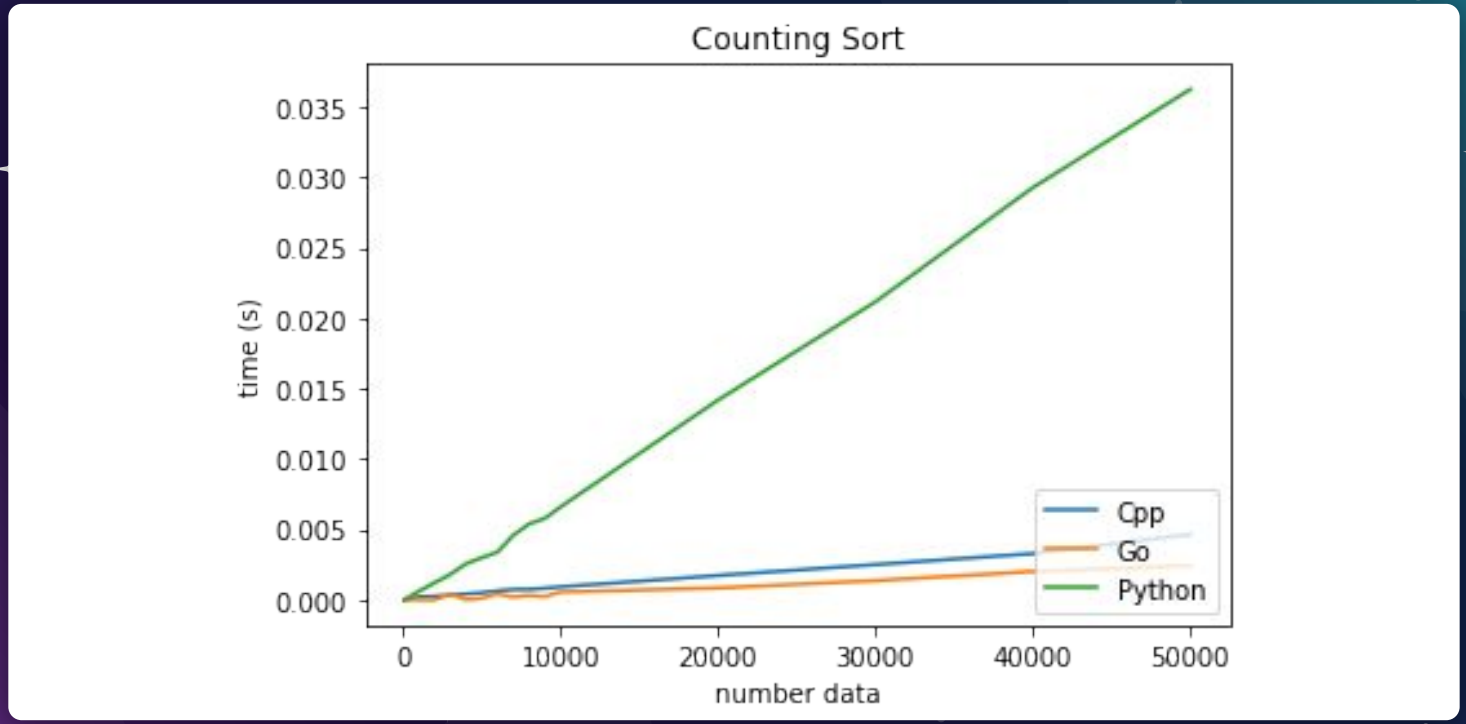
```python
def cocktailSort(list):
  last = len(list) - 1
  while True:
    swapped = False
    for i in range (last-1,1,-1):
      if list[i] > list[i+1]:
        list[i], list[i+1] = list[i+1], list[i]
        swapped = True

    if swapped == False: return

    swapped = False
    for i in range(last-1,-1,-1):
      if list[i] > list[i+1]:
        list[i], list[i+1] = list[i+1], list[i]
        swapped = True

    if swapped == False: return
```

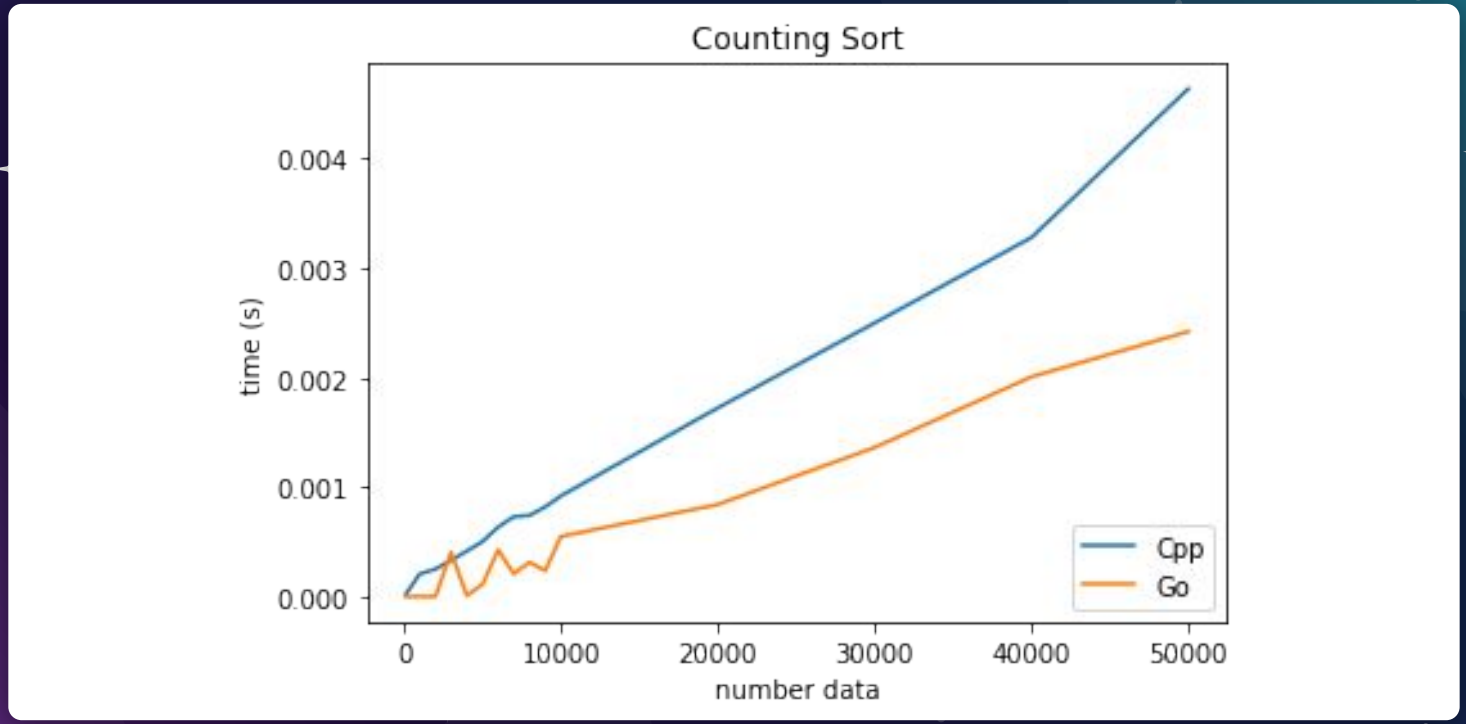
```python
def CountingSort(list):
  minItem = min(list)
  countList = [0] * (max(list) - minItem + 1)
  outputList = [0] * len(list)

  for i in range(0, len(list)): countList[list[i] - minItem] += 1

  for i in range(1, len(countList)): countList[i] += countList[i-1]

  for i in range(len(list)-1, -1, -1):
    outputList[countList[list[i] - minItem] - 1] = list[i]
    countList[list[i] - minItem] -= 1

  for i in range(0, len(list)): list[i] = outputList[i]
```

Comparación


Cocktail Sort
time (s)
number data
0
50
100
150
200
250
0
10000
20000
30000
40000
50000
Cpp
Go
Python


Cocktail Sort
time (s)
number data
0
2
4
6
8
10
12
0
10000
20000
30000
40000
50000
Cpp
Go


Counting Sort
time (s)
0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
0
10000
20000
30000
40000
50000
number data
Cpp
Go
Python


Counting Sort
time (s)
0.004
0.003
0.002
0.001
0.000
0
10000
20000
30000
40000
50000
number data
Cpp
Go

Counting Sort Go							
N	1	2	3	4	5	AVG	SD
100	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.000000000
1000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.000000000
2000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.000000000
3000	0.00100780	0.00050640	0.00000000	0.00000000	0.00050650	0.00040414	0.000421900
4000	0.00004710	0.00000000	0.00000000	0.00000000	0.00000000	0.00000942	0.000021064
5000	0.00000000	0.00000000	0.00000000	0.00054790	0.00000000	0.00010958	0.000245028
6000	0.00053600	0.00107970	0.00051180	0.00000000	0.00000000	0.00042550	0.000449927
7000	0.00051750	0.00000000	0.00000000	0.00000000	0.00051670	0.00020684	0.000283227
8000	0.00000000	0.00053940	0.00050740	0.00000000	0.00051680	0.00031272	0.000285710
9000	0.00000000	0.00000000	0.00062060	0.00055750	0.00000000	0.00023562	0.000323406
10000	0.00051380	0.00055530	0.00013060	0.00050420	0.00104120	0.00054902	0.000324270
20000	0.00051370	0.00108970	0.00101970	0.00099720	0.00058210	0.00084048	0.000270341
30000	0.00158390	0.00161050	0.00157420	0.00099800	0.00104110	0.00136154	0.000312847
40000	0.00216950	0.00223260	0.00153590	0.00199770	0.00210200	0.00200754	0.000277645
50000	0.00322110	0.00267620	0.00206540	0.00199480	0.00214750	0.00242100	0.000521581

Conclusión


FIN