Bubble vs Insertion Algorithms

Max Maldonado
Universidad de Artes Digitales

Guadalajara, Jalisco

Email: idv16a.msolano@uartesdidgitales.edu.mx

Profesor: Efraín Padilla

Mayo 09, 2019

I. Definición del Problema

En éste ejercicio comparamos dos algoritmos de ordenamiento de valores enteros. Los algoritmos que se estudian en este documento son Bubble e Insertion. El problema reside en la necesidad de observar el rendemiento de ambos algoritmos en tres casos específicos: el mejor caso, el peor caso y el caso promedio. El mejor caso radica en introducir un vector ya ordenado dentro del algoritmo. El peor caso radica en introducir un vector ordenado de forma inversa dentro del algoritmo. El caso promedio radica en introducir un vector de orden aleatorio dentro del algoritmo.

II. Inputs y Outputs

Input	Lista de N números enteros cuyo orden es desconocido.
Output	Lista de N números enteros ordenados de ascendentemente.

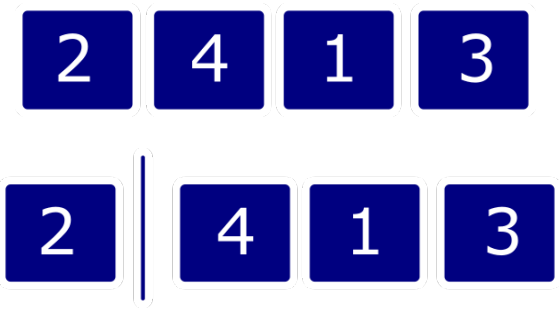
III. Solución

A. *Bubble Sorting*

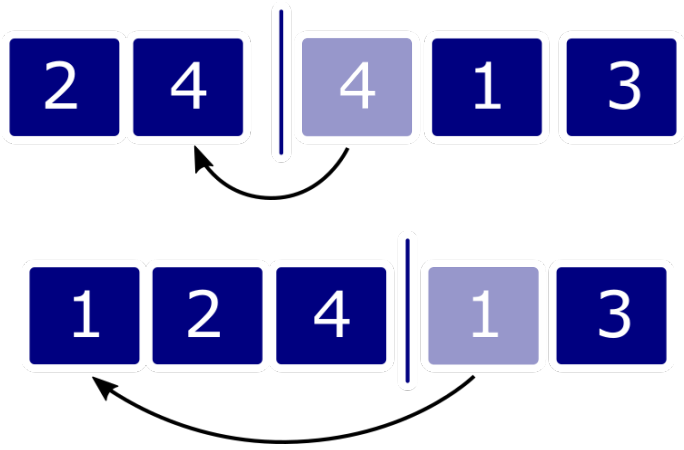
Este algoritmo realiza comparaciones entre los elementos de la lista. Este algoritmo tiene un comportamiento N al cuadrado. Este algoritmo compara cada elemento con todos los demás elementos de la lista, para posiciónarlo correctamente. Debido a que cada debe recorrer cada elemento.

B. *Insertion Sorting*

Éste algoritmo divide nuestro lista de elementos en dos. Una división contiene el primer elemento y la otra división contiene el resto de elementos. Hasta este punto podemos decir que " la primera divisón está correctamente ordenada ", pues sólo tiene un elemento.



A partir de este punto, comparamos los elementos de la derecha con los elementos de la izquierda para ordenarlos correctamente.



IV. Best / Worst Complexity

A. Best

El mejor escenario consiste en una cadena de números que ya se encuentran ordenados (Ascendente).

B. Worst

El peor escenario consiste en una cadena de números que está ordenada de manera opuesta (Descendente).

V. Code

A. Bubble Sorting

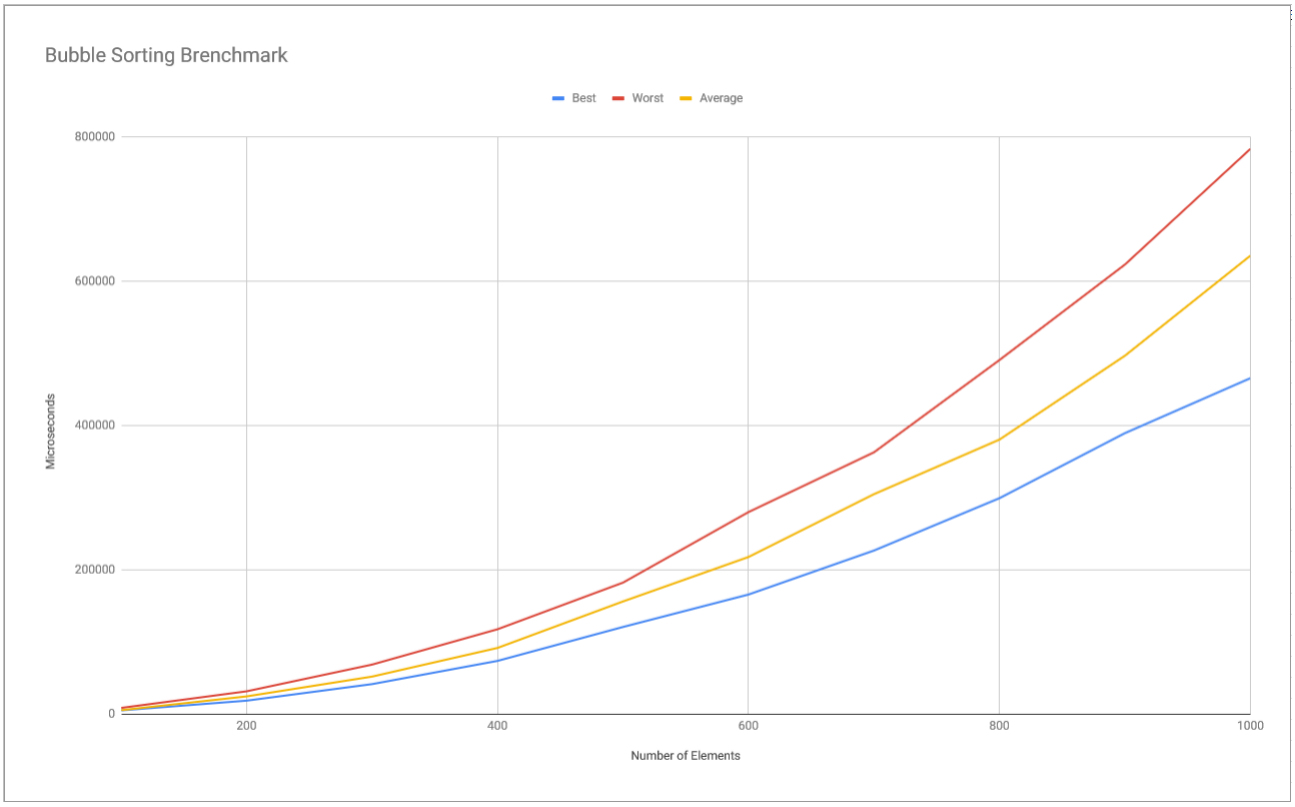
```
void
bubbleAscSorting(std::vector<int> & _vector)
{
  int vecSize = _vector.size();
  int vecSizeR = vecSize - 1;

  for (int indexPass = 0; indexPass < vecSize; ++indexPass)
  {
    for (int idxPos = 0; idxPos < vecSizeR; ++idxPos)
    {
      if (_vector[idxPos] > _vector[idxPos + 1])
      {
        std::swap(_vector.at(idxPos), _vector.at(idxPos + 1));
      }
    }
  }
}
```

B. Insertion Sorting

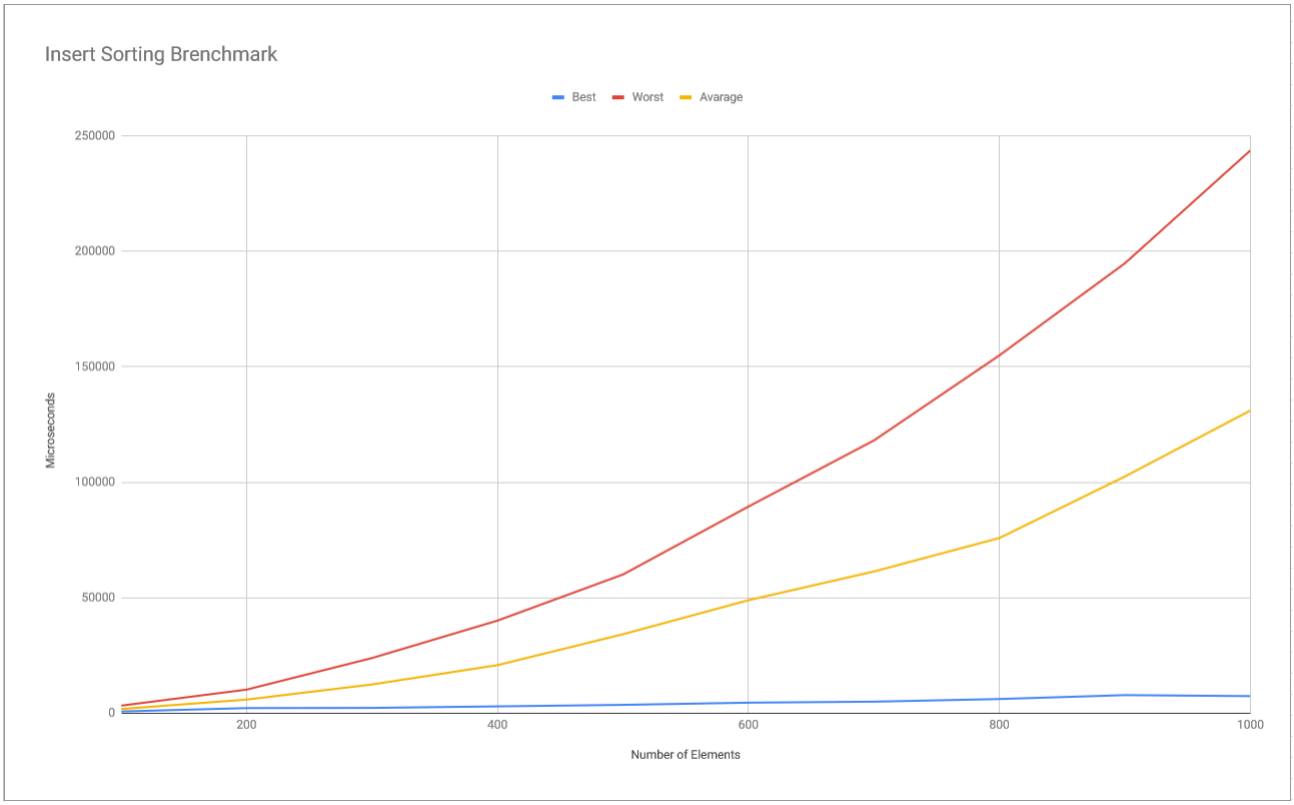
```
void
insertionAscSorting(std::vector<int> & _vector)
{
  if (_vector.size() < 2)
  {
    return;
  }

  int anchorPosition  = 1;
  int itPosition      = 0;
```

```
  int vecSize          = _vector.size();

  while (anchorPosition < vecSize)
  {
    // Check values before anchor position
    while (itPosition >= 0)
    {
      if (_vector[itPosition] < _vector[anchorPosition])
      {

        // Move element
        _vector.insert(_vector.begin() + itPosition + 1, _vector[anchorPosition]);
        _vector.erase(_vector.begin() + anchorPosition + 1);

        break;
      }

      if (itPosition == 0)
      {
        // Move element
        _vector.insert(_vector.begin(), _vector[anchorPosition]);
        _vector.erase(_vector.begin() + anchorPosition + 1);
      }

      itPosition--;
    }

    itPosition = anchorPosition;
    anchorPosition++;
  }

  return;
}
```

VI. BrenchMark


Bubble Sorting Brenchmark
Best
Worst
Average
800000
600000
400000
200000
0
Microseconds
200
400
600
800
1000
Number of Elements

Bubble Sorting			
Elements	Best	Worst	Average
100	5099	8547	5392
200	18603	31561	24528
300	41537	68610	51966
400	73751	117623	91726
500	120779	182265	156155
600	165744	280026	217776
700	226714	362872	304849
800	299276	490791	380501
900	389425	623140	496760
1000	465592	783482	635450


Insert Sorting Brenchmark
Best
Worst
Avarage
250000
200000
150000
100000
50000
0
Microseconds
200
400
600
800
1000
Number of Elements

Insert Sorting			
Elements	Best	Worst	Average
100	698	3224	1767
200	2157	10166	5846
300	2244	23852	12405
400	2903	40076	20781
500	3558	60051	34166
600	4508	89484	48906
700	4944	118116	61347
800	6090	154974	75846
900	7775	194852	102490
1000	7351	243746	131067