

Análisis y Algoritmos

Sergio Juan Díaz Carmona
Universidad de Artes Digitales

Guadalajara, Jalisco

Email: idv17c.sdiaz@uartesdigitales.edu.mx

Profesor: Efraín Padilla

Mayo 5, 2019

1) Algoritmos de Ordenamiento

En computación y matemáticas un algoritmo de ordenamiento es un algoritmo que pone elementos de una lista o un vector en una secuencia dada por una relación de orden, es decir, el resultado de salida ha de ser una permutación —o reordenamiento— de la entrada que satisfaga la relación de orden dada.

Desarrollamos dos metodos de ordenamientos, "bubble sort" e "insertion sort", ambos tendran las mismas entradas; una entrada en orden acendente, una entrada en orden random y una entrada en orden decendente, todas con la misma cantidad de datos. sin embargo en las tablas solo se tendran en cuenta el mejor y el peor caso, que serian la acendente y la decenten, en ese orden. en ambos casos el tamaño de la entrada sera en "n"

BUBBLE SORT

La Ordenación de burbuja es un sencillo algoritmo de ordenamiento. Funciona revisando cada elemento de la lista que va a ser ordenada con el siguiente, intercambiándolos de posición si están en el orden equivocado.

En el caso de Bubble sort el mejor caso (omega) es de $(n \text{ a la } 2)$, mientras que el peor caso (OH) es de $(n \text{ a la } 3)$

Esto es debido a que tiene que recorrer n veces los n numeros, pero al ser en orden acendente no realiza nada mas que la revision,

por otro lado, en el peor caso, la revision es lo mismo, sin embargo se realizan n cambios
el codigo es el siguiente

```
std::vector<int> ordenamientobubble(std::vector<int> vec)
{
    for (int i = 1; i < vec.size(); i++)
    {
        for (int j= 0; j < vec.size()-i; j++)
        {
            if (vec[j]>vec[j+1])
            {
                int num = vec[j];
                vec[j] = vec[j + 1];
                vec[j + 1] = num;
            }
        }
    }
    return vec;
}
```

INSERTIO SORT

El ordenamiento por inserción (insertion sort en inglés) es una manera muy natural de ordenar para un ser humano, y puede usarse fácilmente para ordenar un mazo de cartas numeradas en forma arbitraria.

El mejor caso de insertio (omega) es de $(n)(n-1)$ debido a que revisa uno por uno pero mientras va ordenando por los que ya paso, sin embargo el peor caso es que tenga que acomodar todos (oh) que seria el caso de $(n \text{ a la } 2)$

el codigo es el siguiente

```

std::vector<int> insertion(std::vector<int> vec)
{
    int j;
    int actual;
    int size = 1;
    for (int i = 0; i < vec.size(); i++) {
        actual = vec[i];
        for (j = i; j > 0 && vec[j - 1] > actual; j--) {
            vec[j] = vec[j - 1];
        }
        vec[j] = actual;
        size++;
    }
    return vec;
}

```

A continuacion se muestran las graficas de ambos casos, ambas con el mejor caso (ascendente) y el peor caso (descendente).

