Algoritmos de ordenamiento

¿Por qué ordenar?

Conceptos

- Estabilidad: Se mantiene el orden inicial en caso de empates.

- Adaptabilidad: Son afectados por el preordenamiento de la entrada.

- Algoritmos basados en comparación.

Basados en comparaciòn.

- $O(\ nlogn\ )$

Heap Sort
Merge Sort
QuickSort

- $O(\ n^2\ )$

Burbuja
Inserción
Selección

Quick Sort

Tiempo de ejecución esperado: $\Theta(nlogn)$.

Utiliza el paradigma de divide y vencerás.

Particularmente rápido en el caso promedio.

Quick Sort - D&C

Considerando un arreglo *A[p .. r]*:

Divide: Particionar el arreglo en dos subarreglos *A[p..q-1]* y *A[q+1..r]* tal que los elementos del primer subarreglo son menores o iguales que A[q], que a su vez es menor o igual a los elementos del segundo.

Conquista: Ordenar *A[p..q-1]* y *A[q+1..r]* de manera recursiva.

Combina: No necesario, ya que el arreglo esta ordenado.

Quick Sort - Pseudocódigo

```
QUICKSORT ( A , p , r )
    if p < r
    q = PARTITION( A , p , r )
    QUICKSORT( A , p , q - 1 )
    QUICKSORT( A , q + 1 , r )
```

Quick Sort - Pseudocódigo

```
PARTITION( A , p , r )
        x = A[ r ]
        i = p - 1
        for j = p to r-1
                if ( A[ j ] ≤ x )
                        i = i + 1
                        exchange A[ i ] with A[ j ]
        exchange A[ i+1 ] with A[ r ]
        return i + 1
```

Quick Sort

$Si\ p \leq k \leq i,\ entonces\ A[\ k\ ] \leq x.$

$Si\ i + 1 \leq k \leq j - 1\ ,\ entonces\ A[\ k\ ] > x.$

$Si\ k = r\ ,\ entonces\ A[\ k\ ] = x.$

Quick Sort - Desempeño

Depende de que tan balanceado sea el particionamiento (orden relativo de la entrada) , entre más balanceado mejor tiempo de ejecución, aún así el tiempo promedio se acerca más al mejor tiempo que al peor.

Porque no es lo mismo…

“Dada una entrada random, el tiempo de ejecución esperado del QS es $\Theta(nlogn)$”

que:

“Con alta probabilidad un QS randomizado, se ejecutará en $\Theta(nlogn)$”

Ordenamiento en tiempo lineal

Counting Sort

Se utiliza sólo para entradas contables, asume que la entrada contiene un rango de valores menor al tamaño de la entrada, el tiempo de ejecución viene dado por la diferencia entre el valor mínimo y máximo.

Generalmente es utilizado como apoyo para algoritmos menos limitados.

https://www.hackerrank.com/contests/epiccode/challenges/begin-end

Bucket Sort

Es un algoritmo de distribución, se definen contenedores (cubetas) con propiedades excluyentes de manera que cada elemento sólo puede pertenecer a una de ellas, posteriormente se puede resolver de manera recursiva, u ordenar cada contenedor por separado.
Asume una distribución uniforme.

Radix Sort

Ordenamiento por agrupación, utilizable no sólo para enteros, sino para cadenas, fechas, datos...

Útil a la hora de ordenar enteros (LSD) , o cuando se realiza un ordenamiento lexicográfico (MSD).

Su eficiencia puede ser tan buena o incluso mejor que los ordenamientos basados en comparación.

Comparación de datos no nativos

Comparación de objetos

Sobrecarga de operadores.

```
bool T::operator <(const T2 &b) const;
bool operator <(const T &a, const T2 &b);
```

Librería de c++:

```
#include <algorithm>    // std::sort , std::stable_sort
```

Casos de estudio

Problemas relacionados

Counting Sort
http://codeforces.com/problemset/problem/37/A

Ordenar pares de números
http://codeforces.com/problemset/problem/456/A

Orden Lexicográfico
http://codeforces.com/problemset/problem/515/C

Problemas relacionados II

Sólo ordenar :)
http://codeforces.com/problemset/problem/405/A

¿Por qué es necesario ordenar y qué?.
http://codeforces.com/contest/285/problem/C

Alguna heurística.
http://codeforces.com/problemset/problem/545/D

Material Visual

http://visualgo.net/sorting.html#
http://www.sorting-algorithms.com/
https://www.youtube.com/watch?v=Nz1KZXbghj8
https://www.youtube.com/user/AlgoRythmics

Referencias

Cormen, Thomas H.; Leiserson, Charles E., Rivest, Ronald L., Stein, Clifford (2009) [1990]. **Introduction to Algorithms** (3rd ed.). MIT Press and McGraw-Hill. ISBN 0-262-03384-4.

The Stony Brook Algorithm Repository
http://www3.cs.stonybrook.edu/~algorith/video-lectures/

Sorting algorithm
https://en.wikipedia.org/wiki/Sorting_algorithm