
Algorithms for Problem Solving – 11650

Sorting Algorithms and related problems

Jon Ander Gómez Adrián
`jon@dsic.upv.es`

Departament de Sistemes Informàtics i Computació
Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

9 de febrero de 2014

Algunos algoritmos de ordenación

Algunos
algoritmos de
▷ ordenación
Uso en C
Uso en C++
Uso en Java
Stack of Flapjacks
Bridge I
Bridge II
Bridge III
Shell Sort I
Shell Sort II

Algorithm	Worst case
Selection Sort	$O(n^2)$
Insertion Sort	$O(n^2)$
Shell Sort	$O(n \log^2 n)$
Quick Sort	$O(n \log n)$
Merge Sort	$O(n \log n)$
Heap Sort	$O(n \log n)$
Counting Sort	$O(n)$

Algunos algoritmos
de ordenación

▷ Uso en C

Uso en C++

Uso en Java

Stack of Flapjacks

Bridge I

Bridge II

Bridge III

Shell Sort I

Shell Sort II

- `qsort()`
- `bsearch()`
- Función de comparación para ambas funciones
 - ver código fuente `sorting.c` –

Uso de la STL en C++

Algunos algoritmos
de ordenación

Uso en C

▷ Uso en C++

Uso en Java

Stack of Flapjacks

Bridge I

Bridge II

Bridge III

Shell Sort I

Shell Sort II

- `sort()`
- `stable_sort()`
- `binary_search()`
- Función de comparación para ambas funciones
- Clase que implemente el operador `()` para comparar.

– ver código fuente `sorting_stl.cpp` –

Algunos algoritmos
de ordenación

Uso en C

Uso en C++

▷ Uso en Java

Stack of Flapjacks

Bridge I

Bridge II

Bridge III

Shell Sort I

Shell Sort II

- Interfaz Comparable<T>
- Interfaz Comparator<T> y ejemplo de una clase que la implementa
- Arrays.sort(...)
- Arrays.binarySearch(...)
- Ver código en:
 - ObjectComparator.java
 - CheckingSortingAlgorithms.java

Problema 110402/120

- Intentar ordenar estilo *selección directa*.
- Debemos dar un volteo para colocar el más grande de los que quedan en la última posición, si no lo está ya.
- Debemos dar otro volteo para dejarlo en la posición que le corresponde.
- Los índices de uno y otro volteo han de calcularse adecuadamente.

Hay disponible código sin solución que carga la entrada.

Problema 110403/10037

- En primer lugar ordenar de más rápido a más lento las personas del grupo a pasar (cada caso).
- Una vez ordenadas pasaremos las más lentas combinando con las más rápidas para que devuelvan la linterna.
- Cuando quede una persona por pasar no representa ningún problema. Si son dos tampoco.
- En el caso de que tengamos 4 personas: A, B, C y D.
¿Cuántas maneras de pasarlos posibles tenemos?
¿Qué estrategia conviene?
- ¿Y si solo quedan 3?

Bridge II

Algunos algoritmos
de ordenación

Uso en C

Uso en C++

Uso en Java

Stack of Flapjacks

Bridge I

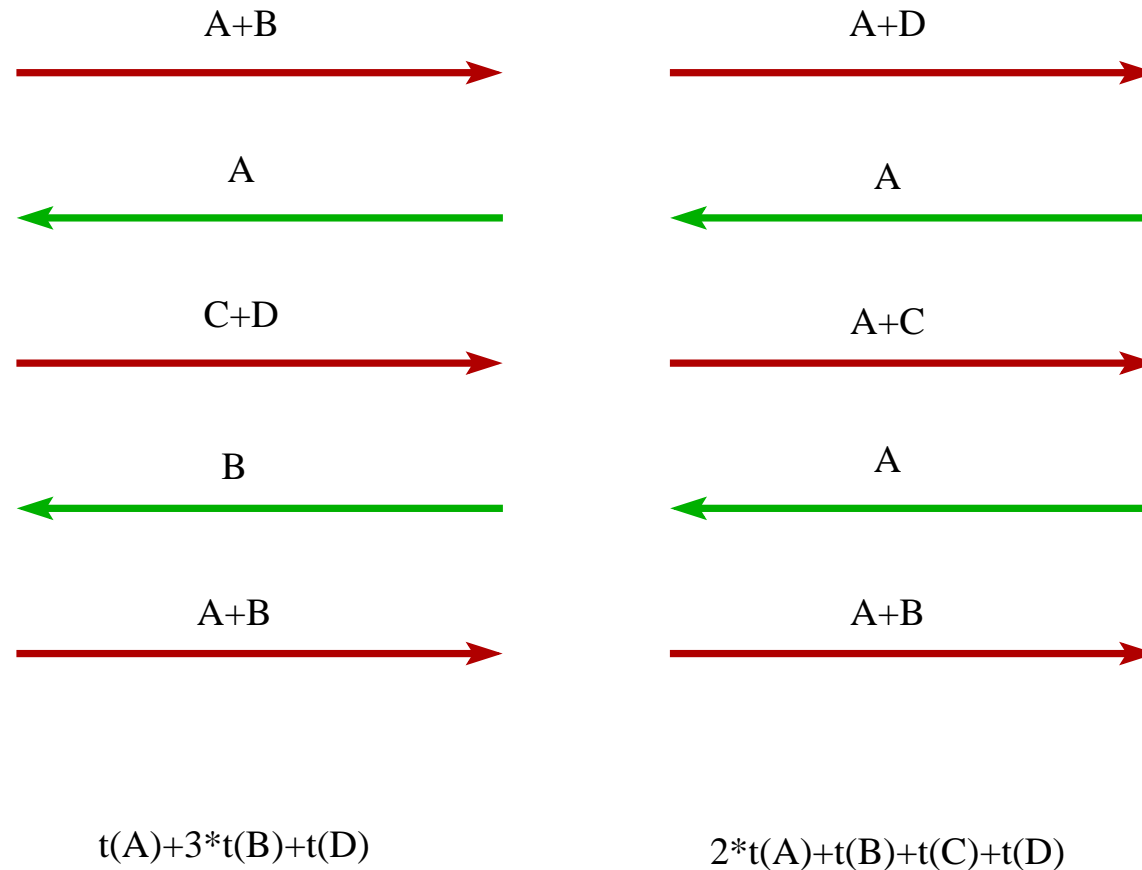
▷ Bridge II

Bridge III

Shell Sort I

Shell Sort II

Problema 110403/10037



Algunos algoritmos
de ordenación

Uso en C

Uso en C++

Uso en Java

Stack of Flapjacks

Bridge I

Bridge II

▷ Bridge III

Shell Sort I

Shell Sort II

Problema 110403/10037

Para este problema el procesar la entrada no tiene ninguna compliación.

```
...
int P[1000];
...
scanf( "%d", &numCasos );
fgets( linia, 120, stdin );
while( --numCasos >= 0 ) {
    scanf( "%d", &n );
    for( i=0; i <n; i++ ) scanf( "%d", &P[i] );
    /* Aquí solucionar cada caso */
}
```

Además del vector P con los tiempos será necesario mantener más estructuras de datos.

Algunos algoritmos
de ordenación

Uso en C

Uso en C++

Uso en Java

Stack of Flapjacks

Bridge I

Bridge II

Bridge III

▷ Shell Sort I

Shell Sort II

Problema 110407/10152

- No confundir el nombre del problema con el algoritmo de ordenación
- Por cada nombre guardar
 - la posición que ocupa en el listado original
 - y la que ocupa en el listado definitivo
- Ordenando de manera ascendente según la posición definitiva podemos observar ...

Shell Sort II

Algunos algoritmos
de ordenación

Uso en C

Uso en C++

Uso en Java

Stack of Flapjacks

Bridge I

Bridge II

Bridge III

Shell Sort I

▷ Shell Sort II

Problema 110407/10152

Duke of Earl	1	0
Yertle	0	1
Sir Lancelot	2	2
<hr/>		
Duke of Earl		

Yertle	0	0
Richard M. Nixon	5	1
Sir Lancelot	2	2
Duke of Earl	1	3
Elizabeth Windsor	3	4
Michael Eisner	4	5
Mr. Rogers	6	6
Ford Perfect	7	7
Mack	8	8
<hr/>		

Sir Lancelot
Richard M. Nixon
Yertle