

## Fondamenti di informatica II (1° canale)

### Esercitazione in Laboratorio del 31 marzo 2020

#### Sorting

L'esercitazione si propone di realizzare, testare e confrontare alcuni algoritmi di *sorting*, su vari input generati dal driver che viene messo a disposizione. Gli algoritmi che possono essere realizzati e testati includono:

- mergeSort
- heapSort
- selectionSort
- insertionSort
- quickSort
- radixSort
- bucketSort

ma il driver può essere facilmente modificato per testarne altri. *Nel driver è già implementato un bubbleSort; nel caso lo si volesse usare occorre "scommentare" in types.h il suo prototipo.*

Gli input per gli ordinamenti possono essere generati randomicamente dal driver, scegliendone la dimensione (<size\_input>, dimensione minima: MIN\_SIZE; dimensione massima: MAX\_SIZE)) e la tipologia (<tipo\_input>):

- C (ordinato crescente),
- D (ordinato decrescente),
- R (random),
- c (quasi-ordinato crescente),
- d (quasi ordinato decrescente).

Il significato è ovvio.

Il driver si usa come di seguito descritto:

- `./driver test <sort_alg> <tipo_input> <size_input>`  
*ai fini della correttezza: esegue <sort\_alg> (uno di quelli su elencati) su un input di dimensione <size\_input>, della tipologia <tipo\_input>, stampando info sulla correttezza;*
- `./driver run <sort_alg> <tipo_input> <size_input>`  
*esegue ordinamento con <sort\_alg> su un input di tipo <tipo\_input> e dimensione <size\_input>; stampa tempo di elaborazione;*
- `./driver file <sort_alg> <file_name>`  
*come run, ma prendendo input da file;*
- `./driver gen <tipo_input> <size_input> <file_name>`  
*genera un possibile input (tipologia <tipo\_input>, dimensione <size\_input>), scrivendolo nel file <file\_name>;*
- `./driver cmp <nome-alg1> <nome-alg2> <tipoInput> <size>`  
*confronta i due algoritmi <nome-alg1> <nome-alg2>.*

Tutti gli algoritmi di ordinamento hanno signature

```
void <sort_alg>(array *);
```

ove il tipo array è così definito (con ovvio significato della notazione):

```
typedef struct {  
    int *arr;  
    int size;  
} array;
```

L'esercitazione va svolta in C, poiché l'impiego di Java rende più difficoltoso un equo confronto fra gli algoritmi.