

L3. Práctica 3. QuickSort

Práctica: QuickSort

Nueva vez usamos Racket.

En clase se explicó de nuevo los fundamentos de QuickSort. Podrás usar cualquier versión de este algoritmo para procesar el archivo de texto dataQS.txt que contiene 10 mil líneas con textos que representan enteros.

Leerás ese archivo como secuencia de enteros que ordenarás con QuickSort.

Como salida de la práctica mostrarás los primeros 10 y los diez enteros después de ordenar los datos.

```
Welcome to DrRacket, version 7.9 [3m].  
Language: racket/base, with debugging; memory limit: 256 MB.  
Desordenados:  
(2148 9058 7742 3153 6324 609 7628 5469 7017 504)  
Ordenados:  
(1 2 3 4 5 6 7 8 9 10)  
cpu time: 234 real time: 236 gc time: 95
```

Código en la siguiente pagina

```

(time
(define (leer-doc filename)
  (with-input-from-file filename
    (lambda()
      (for/list ([line (in-lines)])
        (string->number line))))))

;Codigo sacado de "https://gist.github.com/lambrospetrou/970bcf188eb6061e1bfbfa9e36c29042"
(define (qsrt iarr lt)
  (cond
    [(< 1 (length iarr))
     (let (
          [pivot (car iarr)]
          [gt (lambda (l r) (not (or (lt l r) (equal? l r))))])
       (append
        (qsrt (filter (lambda (x) (lt x pivot)) iarr) lt)
        (filter (lambda (x) (equal? x pivot)) iarr)
        (qsrt (filter (lambda (x) (gt x pivot)) iarr) lt))])
    [else iarr]))

(define (impr filename p)
  (define (loop i)
    (if (= i p)
        '()
        (cons (list-ref filename i) (loop (+ i 1)))))
  (loop 0))

(display "Desordenados:")
(newline)
(display (impr (leer-doc "dataQS.txt") 10))
(newline)
(display "Ordenados:")
(newline)
(display (impr (qsrt (leer-doc "dataQS.txt")<)10))
(newline)
)

```