
Ricorsivo center|400 Fino ad arrivare a vettori da un elemento. Per unire i due subarray ordinati si usa la procedura `merge(A, p, r, q)`. Vengono effettuati circa n confronti, quindi la complessità di merge è $\Theta(n)$. L'idea di merge si basa su: - Copiare temporaneamente `A[p:r]` e `A[r+1:q]` in due array L e R, poi unirli in `A[p:q]`.

[!attention]- Code

```
1 mergesort(A, p, q){
2   if (p<q){
3       r = (p+q)/2
4       mergesort(A, p, r)
5       mergesort(A, r+1, q)
6       merge(A, p, r, q)
7   }
8 }
9
10
11 // input: A[p:q], A[r+1:q] entrambi ordinati
12 // output: A[p:q] ordinato
13 merge(A, p, r, q){
14     nL = r-p+1 // length of A[p:r]
15     nR = q-r // length of A[r+1: q]
16     L[1:nL], R[1:nR] // initialize two arrays
17     for(i=1 to nL){
18         L[i] = A[p-1 +i]
19     }
20     for(j=1 to nR){
21         R[j] = A[r+j-1]
22     }
23     i = 1, j = 1, k = p
24     while(i<nL and j<nR){
25         if L[i] <= R[j]{
26             A[k] = L[i]
27             i++
28         } else {
29             A[k] = R[j]
30             j++
31         }
32         k++
33     }
34     while(i<nL){
35         A[k] = L[i]
36         i++, k++
37     }
38     while(j<nR){
39         A[k] = R[j]
40         j++, k++
41     }
42 }
```

Correttezza

MergeSort termina sempre con $A[p:q]$ ordinato.

Dimostrazione su $n = q - p + 1 =$ lunghezza porzione da ordinare. BASE: $n=1 \rightarrow A[p:q]$ ha un solo elemento, quindi è ordinato. HPind: MergeSort(A, p, q) termina con $A[p:q]$ ordinato se $q - p + 1 < n$
TSind: MergeSort(A, p, q) termina con $A[p:q]$ ordinato se $q - p + 1 = n$

Dimostrazione: (guardando il codice) Siamo fuori dal caso base quindi $n > 1$, si entra nel corpo dell'if, viene calcolato r . -mergeSort(A, p, r) sapendo che $r - p + 1 < q - p + 1 = n$ per ipotesi induttiva termina con $A[p:r]$ ordinato. -mergeSort($A, r+1, q$) sapendo che $q - (r+1) + 1 < q - p + 1 = n$ per ipotesi induttiva di nuovo termina con $A[r+1, q]$ ordinato.

Si sono verificate quindi le due condizioni necessarie per la funzione merge(), che è corretta, quindi la procedura mergeSort() è corretta.

Complessità

$$\Theta(n \log n)$$

\$