

Nota per il corso di TSD di

ott 2021

Statistiche per Big Data

Algorithm Partition

A lezione abbiamo visto il seguente algoritmo che
in input una lista L e due indici l, h restituisce
un indice i che partiziona la sottolista $L[l:h]$
in

$L[l:i]$ che contiene elementi $< L[i]$
 $L[i+1:h]$ " " " " $> L[i]$

Partition(A, l, h)

1. $i = l - 1$
2. $p = L[h - 1]$
3. for j in range(l, h):
4. if $L[j] \leq p$:
5. scambia($L[i+1], L[j]$)
6. $i \leftarrow i + 1$
7. return i

Assumiamo che $l < h - 1$ e che quindi la sottolista
 $A[l:h]$ contenga almeno due elementi

Altrimenti aggiungere

[0. if $l == h - 1$;
 return l]

Dimostriamo che alla fine del ciclo for per $j = l, \dots, h-2$ abbiamo che

(*) $L[l:i+1]$ contiene solo elementi $< p$

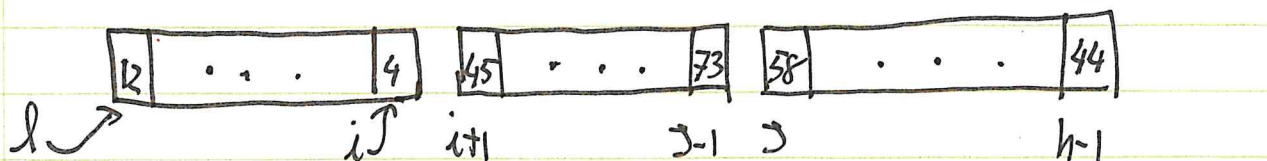
$L[i+1:j+1]$ contiene solo elementi $> p$

e la sottolista $L[j+1:h]$ non è stata ancora esaminata

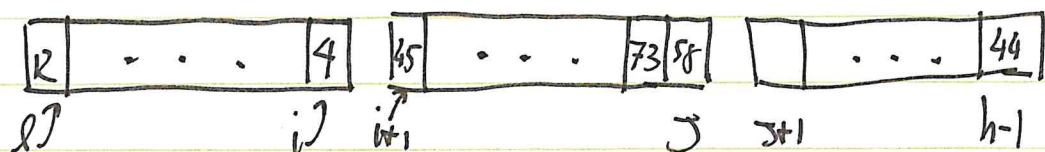
Durante l'esecuzione del j -esimo ciclo due casi sono possi:

1) $L[j] > p$

Ad esempio

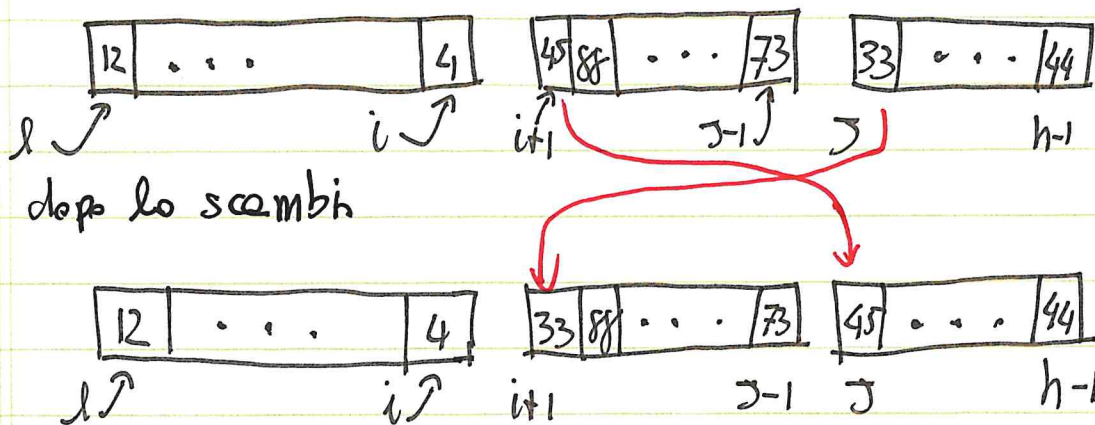


In questo caso l'algoritmo non esegue alcuna operazione e quindi il for incrementa il valore di j , portando alla situazione seguente

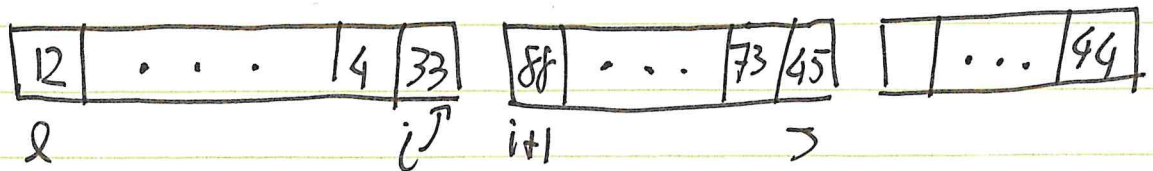


2) $L[j] < P$

In quest caso l'algoritmo prima scambia $L[i+1], L[j]$
 Ad esempio \uparrow passo 5



e successivamente incrementa i (passo 6)
 e j (per effetto del ciclo for)



Quindi se le condizioni (*) sono soddisfatte all'inizio di una iterazione del ciclo for, allora restano soddisfatte alla fine dell'iterazione

Questo vale per $j=l, l+1, \dots, h-1$

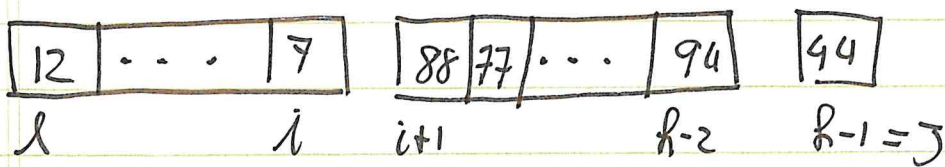
Guardiamo adesso all'ultima iterazione del ciclo for per $j = h-1$

In questo caso $L[j] = L[h-1] = p$ e quindi

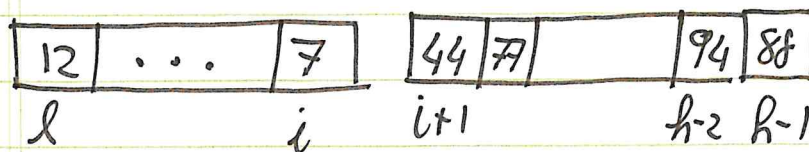
$L[j]$ è scambiato con $L[i+1]$ e poi i è posta uguale a $i+1$

Il ciclo for termina e il valore di i è restituito

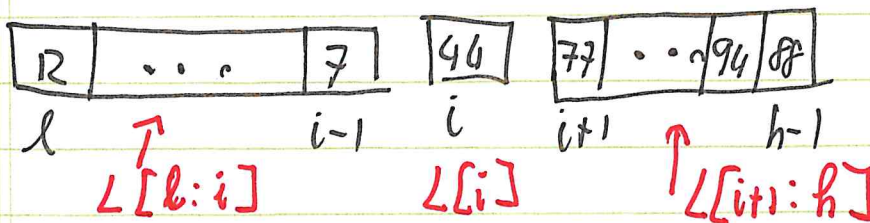
Ad esempio



dopo lo scambio



incrementiamo i



Quindi

$L[l:i]$ contiene elementi $< L[i]$

$L[i+1:h]$ contiene elementi $> L[i]$