

DETERMINAZIONE DEL MASSIMO E DEL SECONDO MASSIMO DI n OGGETTI?

(Gli oggetti sono tutti diversi)

Se si ragiona in modo analogo al caso della ricerca del massimo e del minimo, si può vedere che si ottiene lo stesso risultato.

Con 2 oggetti 1 passo

Con 3 oggetti 1+ 2 passi

infatti dopo aver confrontato due devo confrontare il terzo sia con uno che con l'altro

Con 4 oggetti 1+ 3 passi

infatti dopo aver fatto due gruppi A1 A2 A3 A4 di due e averli confrontati a due a due, confronto i due massimi, ad esempio A1 e A3 e trovo il Massimo, confronto inoltre il più piccolo dei due massimi, ad esempio A3 con A2, cioè con l'elemento con il quale A3 non è stato ancora confrontato, e trovo così il secondo massimo

Per contare il numero dei passi quando gli oggetti sono n, procedo come segue:

Se il numero è pari

$$1 + 3 * (n/2 - 1) = 3 * n/2 - 2$$

Se il numero è dispari

$$3 * (n - 1) / 2 - 2 + 2 = 3 * (n - 1) / 2$$

Identico risultato che nel caso del massimo e minimo

Se invece si fa il seguente calcolo

Cioè 1 passo dovuto al confronto dei primi due, più 2 passi per ogni elemento aggiunto, gli elementi rimasti sono proprio n-2, ottengo

$$1 + 2*(n-2) = 2*n - 3$$

Il risultato, $2*n-3$, si poteva anche raggiungere in altro modo, ragionando come segue:

per trovare il massimo di n elementi, $n-1$ passi,

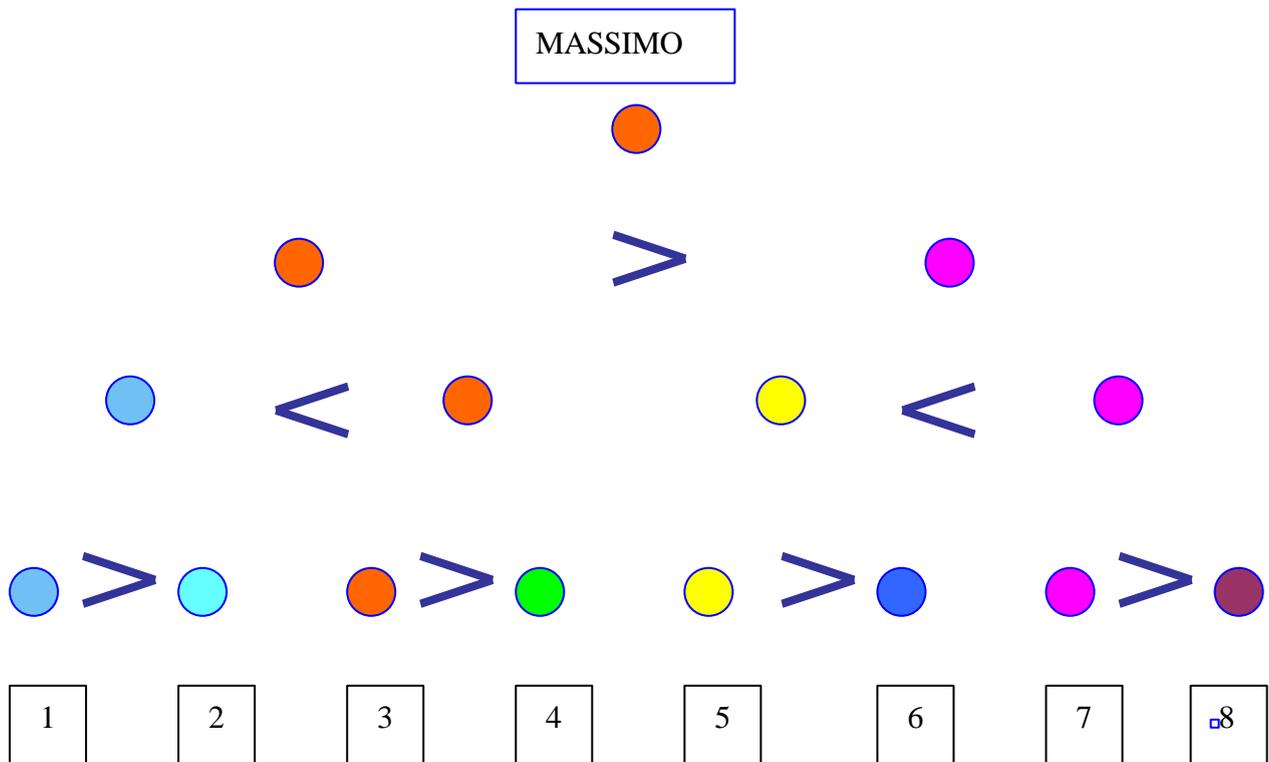
per trovare il massimo di $n-1$ elementi, $n-2$ passi;

Quindi

$$n-1 + n-2 = 2*n - 3$$

$2*n-3$ passi

L'ALGORITMO MIGLIORE PER LA RICERCA DEL I E DEL II DI UN INSIEME



Si può notare che il numero di livelli dell'albero è il numero di divisioni per due del numero di oggetti, quindi, in questo caso 3, che è anche l'esponente da dare a 2 per ottenere 8, cioè $n = 2*2*2$.

Alla fine si vede che l'elemento n.° 3 è il massimo dell'insieme.

Per trovare il massimo dell'insieme ci sono voluti nel nostro caso 7 confronti, in generale $n-1$ confronti.

Come fare per trovare il secondo massimo?

L'albero mette bene in evidenza come il secondo massimo non può che essere uno tra quelli che hanno perso con il primo massimo: la pallina azzurra, la pallina rosa e la verde

Tutte le altre palline diverse dal vincitore rosso hanno perso o con quella rosa o con quella azzurra, e sono quindi certamente sconfitte, non possono certamente essere il secondo massimo, non portano più alcuna informazione.

Invece quelle che hanno perso solo con il vincitore rosso, non hanno veramente perso, hanno perso solo la possibilità di accedere al primo posto.

Il numero di quelle che hanno perso con il vincitore è pari al numero di livelli, dato che il vincitore va su di un livello ad ogni vittoria, cioè ad ogni serie di confronti, quindi ad ogni suddivisione per 2. Basterà ora il confronto tra le ultime tre concorrenti per trovare chi vince il secondo posto.

Per trovare la più grande fra 3 servono 2 passi.

Il numero delle palline che hanno perso con il massimo è pari all'esponente che bisogna dare a 2 per ottenere il numero, e cioè $\log_2 n$.

Per trovare poi il massimo tra queste servono $\log_2 n - 1$ passi, dove il $\log_2 n$ è approssimato per eccesso, perché qualora il numero degli oggetti di un livello non sia divisibile per 2, cioè il numero non è una potenza esatta di 2, basterà approssimare per difetto, lasciando da parte un oggetto che verrà confrontato con il primo che resta solo e questo farà aumentare di un livello l'albero. Anche se i resti dovessero essere tanti quanti i livelli, come per il numero 15, ogni livello dal secondo aumenterebbe di un confronto e quindi l'ultimo elemento dovrebbe confrontarsi con un altro e questo farebbe aumentare il numero dei livelli di uno. Ogni resto, escluso l'elemento finale, fa aumentare i confronti.

esempio

$$\log_2 15 = r \qquad 3 < r < 4$$

h rappresenta il numero dei livelli per un numero che è esprimibile come una potenza di 2 ad esponente intero, come $8 = 2^3$ $h = 3$

Tra 9 e 16 il numero di livelli sarà 4 e così via.

CONCLUDENDO

Il numero di confronti sarà pari al numero di passi che servono per trovare il massimo, $n-1$, più il numero di passi che serve per trovare il secondo, e cioè il numero di elementi da confrontare meno 1, cioè $h - 1$:

$$n/2 + n/4 + \dots + 1 + \log_2 n - 1 = n-1 + \log_2 n - 1$$

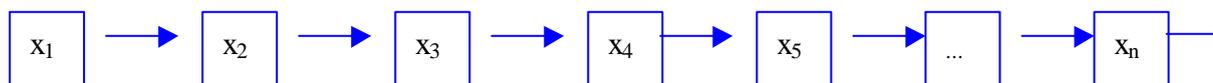
COME COMUNICARE L'ALGORITMO AD UNA MACCHINA?

Per poter sviluppare un programma in Pascal, o in qualche altro linguaggio di programmazione, sono utili le liste, che permettono di memorizzare gli elementi che hanno perduto con il massimo, in modo da non sprecare informazione.

Se si vuole arrivare a un ordinamento totale è necessario conservare ad ogni passo memoria degli sconfitti con uno qualunque degli elementi dell'insieme, perché non sono veramente sconfitti..

Per riuscire a comunicare ad una macchina questo procedimento in modo abbastanza lineare, si può utilizzare *le liste*

Si può visualizzare il procedimento nel modo che segue.

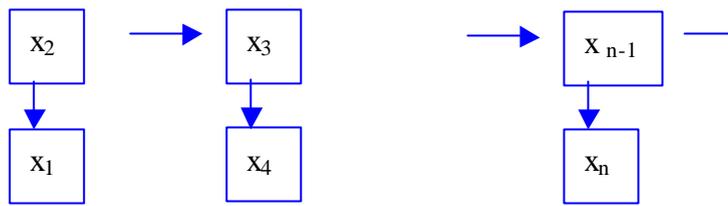


Si confrontano gli elementi della lista a due a due

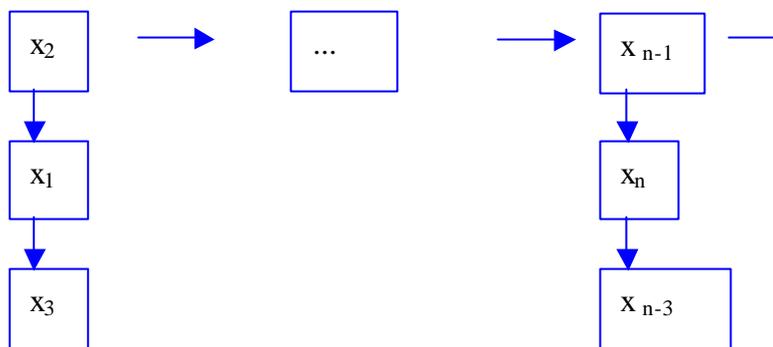
Supponiamo che si abbia

$$x_2 > x_1 \quad x_3 > x_4 \quad \text{e così via}$$

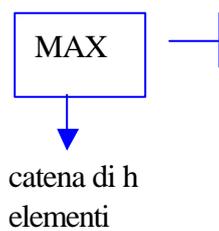
Si può tenere memoria di tutti i confronti con un sistema che si può rappresentare come segue



Continuando si ha



Alla fine resta



h è il numero delle divisioni per due, quindi è l'esponente che bisogna dare a 2 per ottenere il numero, cioè il $\log_2 n$, (è la lunghezza del numero in base 2, diminuita di uno)

Il massimo trovato è seguito dalla catena degli elementi che ha sconfitto, in modo tale da non perdere informazioni e poterle così riutilizzare nel momento in cui bisognerà trovare il secondo dell'insieme.

E se si vuole arrivare a un ordinamento bisogna per ogni elemento dell'insieme tenere conto di tutti gli sconfitti.