ALGORITMI DI ORDINAMENTO

(Unica informazione iniziale sugli insiemi: gli oggetti sono tutti diversi)

Come per l'unità precedente si deve dare la possibilità ai ragazzi di prendere confidenza con il problema, giocando con gli oggetti, ad esempio con le carte, o osservando programmi già pronti che visualizzino l'algoritmo risolutivo.

PRIMI ESEMPI: ORDINAMENTO DI UN PICCOLO NUMERO DI OGGETTI

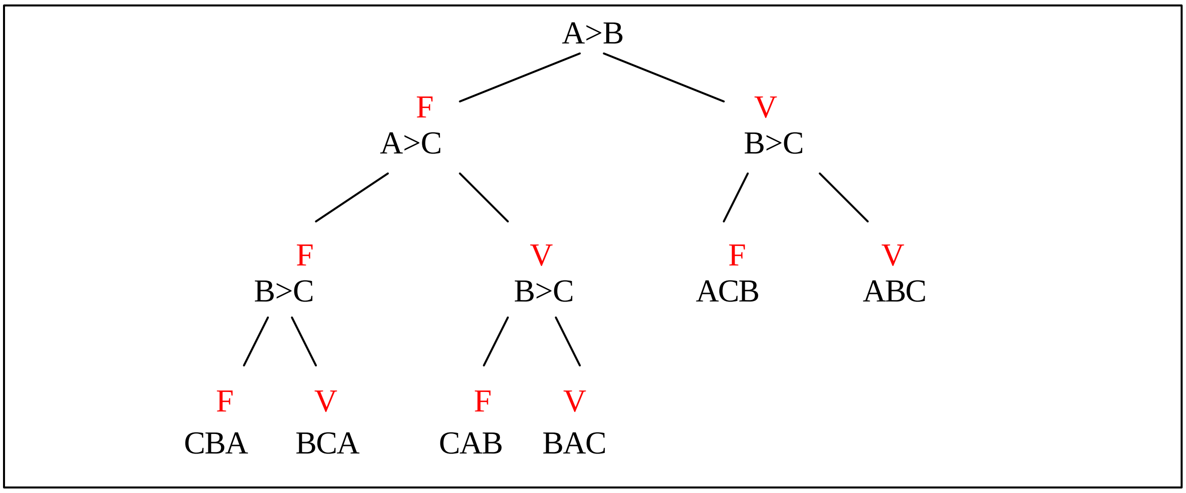
Il numero di oggetti è n con $1 \leq n \leq 5$

Ordinare 2, 3, 4 oggetti, costruendo l'albero binario.

Come si fa a sapere se il numero di passi è ottimale?

Si deve poter ottenere il numero totale di permutazioni degli n oggetti considerati, n!, per determinare tutti i possibili ordinamenti. Si controlla se con un albero che abbia un numero di passi inferiore al numero di passi utilizzato per trovare l'allineamento cercato, posso ottenere un numero di oggetti che sia almeno pari al numero di oggetti che mi serve.

Se il numero di oggetti è uguale o superiore al numero di elementi che mi serve non è però detto che contenga tutte le permutazioni possibili, perché potrebbero esserci alcuni allineamenti ripetuti più volte.



L'ordinamento cercato è uno degli allineamenti possibili che si ottengono permutando gli elementi dell'insieme. La ricerca dell'allineamento voluto avviene tra tutti quelli possibili. Si può pensare quindi di dover effettuare una ricerca in un insieme non ordinato.

Il numero delle permutazioni aumenta in modo sorprendente con l'aumentare del numero n.
Questo fa capire fin dall'inizio quanto ardua sia la ricerca all'aumentare del numero degli elementi.

Si capirà anche con chiarezza l'importanza di strategie opportune per la ricerca e la gestione intelligente dell'informazione acquisita.

Come si è potuto notare nelle pagine precedenti l'informazione si nasconde, alle volte è molto difficile evidenziarla, e inoltre si propaga, come in una reazione a catena, e sfugge facilmente al controllo.

Si suppone che un algoritmo che utilizzi confronti ha sempre un albero che lo rappresenta
Un confronto equivale a un passo, equivale a un livello dell'albero binario che costruisco

2 oggetti	1 passo	$2^1 = 2$	$2! = 2$
3 oggetti	3 passi	$2^3 = 8$	$3! = 6$
4 oggetti	5 passi	$2^5 = 32$	$4! = 24$

1) Primo caso: banale

2) Secondo caso, 3 oggetti: $2^2 = 4$, $3!=6$, non si può con 2 passi.

3) Terzo caso, 4 oggetti: $2^4=16$, $4!=24$, non si può con 4 passi

ORDINAMENTO DI 5 OGGETTI

$2^7=128$, $5!=120$, Forse si può con 7 passi

Primo modo, il peggiore

Se si ordinano 5 oggetti nel modo più banale, utilizzando semplicemente l'informazione acquisita del massimo dell'insieme precedente, ottengo

$n - 1 + n - 2 + n - 3 + n - 4 \ldots + n - (n - 1) = n * (n - 1) - (n - 1)*(1 + n - 1)/2 = n * (n - 1)/2$

Con 3 oggetti bastano 3 passi

Con 4 oggetti ci vogliono 6 passi

Con 5 oggetti ci vogliono 10 passi

Ordino 4 oggetti con 5 passi

Confronto A con B e C con D

Poi confronto tra loro i due massimi e i due minimi

E così trovo il Massimo assoluto e il minimo assoluto

Infine confronto tra loro i due medi

Ordino 5 oggetti con 9 passi

Ordino prima 4 dei 5 oggetti con 5 passi

Inserisco poi il 5

A B C D E

Confronto E con il primo, con il secondo, con il terzo, con il quarto.

In tutto 5 passi + 4 passi = 9 passi

Ordino 5 oggetti con 8 passi

Prima si ordinano 4 oggetti con 5 passi

Poi si confronta il 5 con uno dei due centrali, e si procede verso i maggiori o verso i minori, nel peggiore dei casi ci vogliono 2 confronti

In tutto

5 passi + 3 passi = 8 passi

Ordino 5 oggetti con 8 passi

Si può procedere anche come segue:

Si ordinano 3 oggetti con 3 passi

Si ordinano 2 oggetti con 1 passo

Si fondono i 3 oggetti ordinati con i 2 oggetti ordinati e si ottengono i 5 oggetti ordinati

Per la fusione servono n-1 passi, quindi 4 passi.

In tutto

3passi + 1 passo + 4 passi = 8 passi

È POSSIBILE ORDINARE 5 OGGETTI CON 7 PASSI

ECCO L'ALGORITMO

A B C D E

AB CD E

Si confrontano gli oggetti della prima coppia e gli oggetti della seconda coppia.

Supponiamo che

A > B e C > D

Successivamente confronto i due minori tra loro (si potrebbero confrontare i due maggiori tra loro).

Supponiamo che

B > D

Naturalmente non e' importante il risultato dei confronti, qualunque sia si procede come segue:

Primo risultato significativo:

tre dei cinque, in questo caso ABD, ordinati in 3 mosse, con un informazione in più, C > D

Secondo aspetto importante

a questo punto colloco E nella terna ABD, per farlo mi servono 2 mosse.

Ottengo EABD, AEBD, ABED, ABDE.

Dopo avere collocato E, posso collocare C.

Questa volta sapere che C > D mi fa risparmiare 1 passaggio, dato che è come se dovessi collocare C in una terna, quindi mi bastano 2 mosse.

QUINDI 7 CONFRONTI IN TUTTO.

NOTA BENE

Non è indifferente collocare prima E o prima C

Se avessi collocato C, avrei perduto l'informazione C > D, perché avrei utilizzato 2 passi lo stesso.

RIFLETTIAMO ANCORA SULL'ORDINAMENTO DI 5 OGGETTI.

Dopo l'esperienza dei giochi del torneo dovremmo aver acquisito un modo utile di procedere che potrebbe aiutare a non perdere informazione.

Cerchiamo un algoritmo diverso da quello precedente, ma che permetta ancora di fare 7 confronti per ordinare 5 oggetti.

1. Determiniamo il Massimo

```
A   B        C   D        E
  A            C          E          4 CONFRONTI
        A                 E
                 A
```

Non dobbiamo però perdere alcuna informazione, quindi dobbiamo trarre tutte le conseguenze dai confronti effettuati.

.A>C>D

A>B

A>E

Dato che A è il massimo, è utile oltre al massimo tenere l'unica informazione aggiuntiva C>D

Determiniamo con l'algoritmo già noto il secondo Massimo.

```
B       C       E
    B           E          2 CONFRONTI
          B
```

Possiamo scrivere, dopo 6 confronti, la seguente quaterna ordinata.

ABCD

Nel trovare il secondo, in questo caso, uno di quelli che potevano diventare minimo è stato eliminato.

Per trovare il minimo basta quindi un solo confronto

D E
D

1 CONRONTO

ABECD oppure ABCED ? Dopo 7 confronti il dubbio è rimasto.

Ci vuole l'ottavo confronto nonostante non sia il caso più sfortunato.

.

Non si riesce in 7 passi sfruttando gli algoritmi noti dei giochi sui tornei.

Evidentemente si spreca informazione, ma dove?

Seguendo la precedente strategia nel caso più sfortunato si fanno addirittura 9 passi.

A B C D E
A C E
A E
A
Massimo

B C E
C E
C
Secondo

B D E
B E
B
minimo

Con 8 passi non si è ancora riusciti ad ordinare gli elementi, infatti

A C _ _ B
I II V

Non ho però informazione alcuna su D ed E, se non che debbono stare tra C e B. Devo quindi effettuare ancora un confronto.

Il numero di confronti risulterà

$$4 + 2 + 2 + 1$$

9 passi in totale.

Proviamo in un altro modo ancora.

Con 4 passi trovo il Maggiore e il minore di quattro oggetti.

D B D **A B C D** A C A	verso l'alto salgono i minori, verso il basso i maggiori.

Nessuna informazione aggiuntiva: sappiamo che C non può che stare tra A e D, e così anche B

A C D A B D

M m

A D

Si può collocare E con 2 passi AECD

Si può collocare B con 2 passi, devo confrontarlo sia con E che con C.

Fa lo stesso se collochiamo prima B (con un passo) poi E (con tre passi).

Anche in questo modo i passi non sono 7, ma 8 nel caso più sfortunato.

Siamo riusciti in un solo modo ad ordinare 5 oggetti in sette passi, il primo proposto nelle pagine precedenti.

Qual è l'informazione che è andata perduta con le ultime startegie utilizzate, o che non è stata gestita bene, oppure quali domande potevano avere un effetto migliore?

La risposta è che l'informazione che non è stata gestita bene nell'ultima strategia è l'informazione che potevo utilizzare dopo aver trovato solo il minimo, o solo il massimo.

Dopo aver trovato solo il minimo, ad esempio, sono a conoscenza che la terna ABD è ordinata e che C>D.

D
B D
A B C D

Ma allora è bene inserire il quinto oggetto E a questo punto, perché con due mosse lo inserisco e poi con ancora due mosse inserisco C.

Splendido! 7 mosse in tutto finalmente!

Siamo riusciti così ad intrappolare l'informazione ed a capire meglio il metodo usato nelle pagine precedenti per ordinare i 5 oggetti in sette mosse, a capirne soprattutto "la portata".

Si può aumentare il numero degli oggetti e provare a ottimizzare il numero dei confronti.

Nella tabella seguente tra parentesi tonda è scritto il numero di passi per trovare il I e il II, fuori della parentesi il numero di passi che potrebbero servire per inserire gli altri elementi.

4 oggetti	5 passi	(3 + 1) + 1	$2^4<4!<2^5$	Il log in base 2,
5 oggetti	7 passi	(4 + 2) + 1	$2^6<5!<2^7$	calcolato per eccesso
6 oggetti	10 passi	(5 + 2) + 3	$2^9<6!<2^{10}$	dei numeri 5, 6, 7
7 oggetti	13 passi	(6 + 2) + 5	$2^{12}<7!<2^{13}$	considerati è sempre 3
8 oggetti	16 passi	(7 + 2) + 7	$2^{15}<8!<2^{16}$	

Per **6 oggetti** basterà ordinare prima 5 e poi inserire il sesto in tre passi

Non si può fare meglio di 10 passi.

ABCDE F

Tre passi per confrontare prima F con C e poi con D e con E, o con B e con A.

Non si riesce ad ordinare invece con 10 confronti i sei elementi sfruttando il gioco del torneo. Se trovo il Massimo, il II, il minimo, possono essere necessari 11 passi come nel caso che segue

A B C D E F A C E A E A **M** n –1 passi	B C E C E C **II** [logn] –1 passi	B D F D F D **m** n/2 – 1 passi

Ho solo un' altra informazione, in questo caso E>F, oltre a sapere che gli altri tre elementi vengono dopo i primi due e prima dell'ultimo. Ho bisogno di due passi per inserire B tra E ed F e poi in blocco inserire la terna tra i primi due e l'ultimo. Pur avendo sfruttato tutte le informazioni sono state necessarie 11 mosse

Se poi sempre con 6 oggetti si trova il Massimo, il II, il III e così via fino al penultimo, nel caso che segue ci servono 12 mosse.

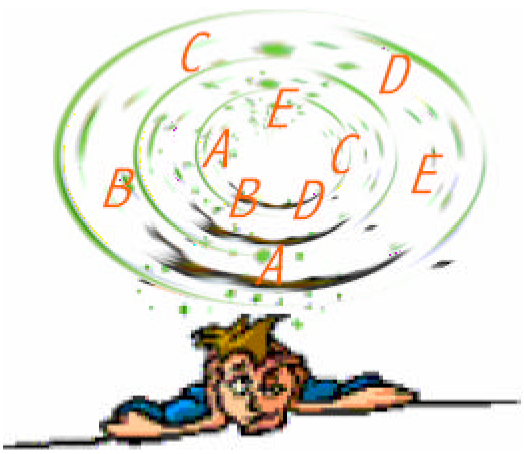
A B C D E F A C E A E A	B C E C E C	E B D E D E	B D F B F B	D F F
M 5 passi	**II** 2 passi	**III** 2 passi	**IV** 2 passi	**V** 1 passo

Si ottiene ACEBFD con 12 mosse. Ma dove si spreca informazione?

Esempio numerico

A	**B**	**C**	**D**	**E**	**F**
50	35	45	25	40	30

Se volessi continuare, per ordinare 7 oggetti, sfruttando l'ordinamento dei 6 con il minor numero di passi possibile e poi inserendo il settimo, mi accorgerei facilmente che non è possibile inserire l'elemento con solo 3 passi, non è quindi possibile seguendo questa strada ordinare 7 oggetti con 13 mosse



La testa mi gira!

Lo scopo è far nascere il problema che segue e sottolinearne la complessità:

I metodi utilizzati per l'ordinamento di un grande numero di oggetti seguono la strada dell'ottimizzazione dei confronti, così come si è provato a fare in queste pagine, o invece si è costretti, proprio in nome dell'efficienza, del minor costo computazionale, a seguire strade diverse?

Se proviamo ad ordinare 5, 6 ... oggetti con uno dei metodi standard, quanti confronti verranno effettuati?

PROVATE PURE...

Si potrà osservare che il numero di confronti sarà superiore al necessario

Potremo confrontare il numero di passi ottenuto con quelli utilizzati in queste pagine per 5, 6 oggetti, di cui conosciamo il minor numero di passi, e successivamente confrontare il numero di passi per ordinare 7, 8, ..., elementi confrontandoli con l'ipotetico numero di passi possibile.

ORDINAMENTO DI UN GRANDE NUMERO DI OGGETTI

Analizzeremo brevemente solo alcuni metodi di ordinamento, anche se sono invece numerose le strategie e gli algoritmi sia per il caso semplice dell'ordinamento di un array, sia per altri casi importanti, nella vita di tutti i giorni, e che richiedano l'ordinamento di sequenze di grosse dimensioni.

In conclusione ci chiediamo se esiste un algoritmo che utilizzi in ogni caso il minor numero di confronti possibile.

Anche se esistesse il procedimento generale con il minor numero di confronti sarebbe conveniente o no utilizzarlo?

È sensato porsi una domanda del genere. Infatti per organizzare tutta l'informazione acquisita è necessario spendere del tempo che potrebbe superare quello guadagnato riducendo i confronti.

1. **:SELECTION SORT**
2. **INSERTION SORT**
3. **MERGE SORT**
4. **QUICK SORT**
5. ...