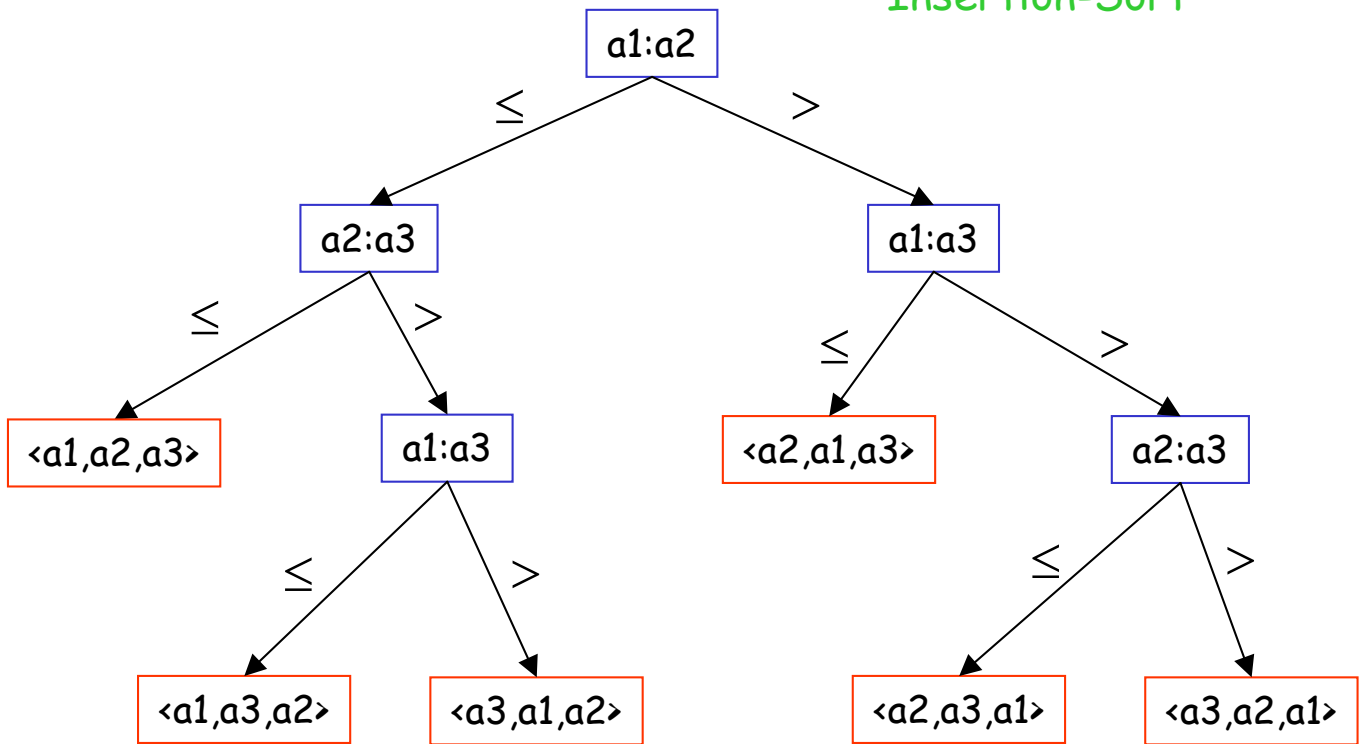
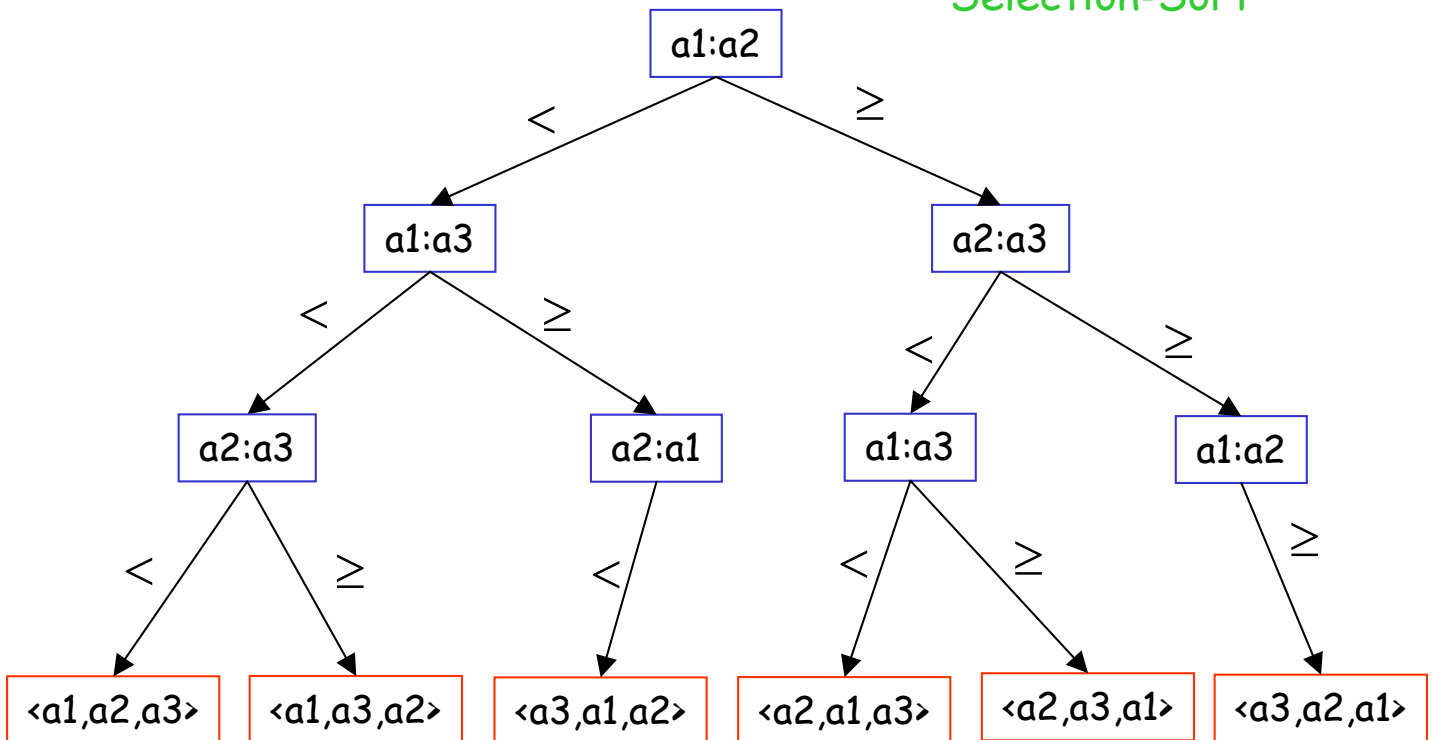


Insertion-Sort



Selection-Sort



Esercizio 2

Consideriamo una funzione $f(n)$ tale che:

$$f(n) = \Omega(n \log_2(n))$$

La relazione precedente implica anche:

$$f(n) = \Omega(n \log_B(n)) \quad B = \text{base generica}$$

Infatti:

$$f(n) = \Omega(n \log_2(n)) \Rightarrow \exists c, n_0 > 0 \text{ tali che } \forall n \geq n_0 \\ 0 \leq c n \log_2(n) \leq f(n)$$

Notiamo che:

$$\log_2(n) = \log_B(n) / \log_B(2)$$

Posso definire $c' = c / \log_B(2)$. Vale allora la relazione:

$$\forall n \geq n_0 \quad 0 \leq c' n \log_B(n) \leq f(n)$$

$$\text{Da cui segue:} \quad f(n) = \Omega(n \log_B(n))$$

Esercizio 3 - Fusione di due sequenze ordinate

Alcune domande preliminari:

- Lower bound ?
- Upper bound ?

```
Merge1(A,B)
For i ← 1 to n/2
    do C[i] ← A[i]
       C[n/2+i] ← B[i]
Insertion-Sort(C)
```

- L'algoritmo e' corretto?
- Complessita' temporale e spaziale dell'algoritmo?
- L'algoritmo e' ottimale?

Algoritmo di fusione

```
Merge2(A,B)
i ← 1
j ← 1
While (i ≤ n/2) and (j ≤ n/2)
    do if (A[i] < B[j])
        then C[i+j-1] ← A[i]
           i ← i + 1
        else C[i+j-1] ← B[j]
           j ← j + 1
If (i > n/2)
    then for k ← j to n/2
        do c[k+n/2] ← B[k]
    else for k ← i to n/2
        do c[k+n/2] ← A[k]
```

- L'algoritmo e' corretto?
- Complessita' temporale e spaziale dell'algoritmo?
- L'algoritmo e' ottimale?
- Cosa succede in termini di confronti ?