

Divide et Impera

- Divisione di un problema in **N sottoproblemi piu piccoli**
- Ricorsione : si **ferma** quando un sottoproblema raggiunge la **dimensione minima**
- Aggregare le soluzioni dei sottoproblemi per formare la **soluzione finale**
- Dimensione minima: quella che permette di risolvere il problema **direttamente**

Codice

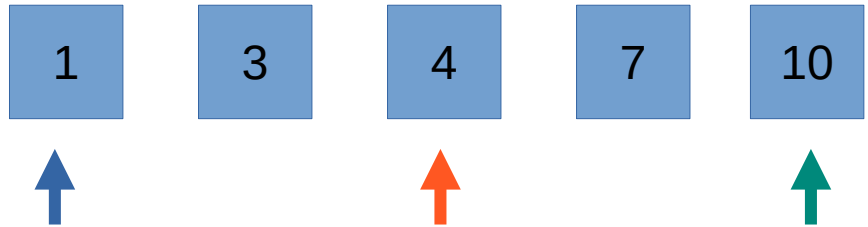
```
def divide_et_impera(problema):  
    if dimensione(problema) <= soglia:  
        return risolvi_direttamente(problema)  
  
    problema1, problema2 = dividi(problema)  
  
    risultato1 = divide_et_impera(problema1)  
    risultato2 = divide_et_impera(problema2)  
  
    return combina(risultato1, risultato2)
```

Ricerca Binaria

```
def cerca(list, n, start, end):  
    if start > end:  
        return -1  
    medio = (start + end)//2  
    if list[medio] == n:  
        return medio  
    elif list[medio] < n:  
        return cerca (list, n, medio+1, end)  
    else :  
        return cerca (list, n, start, medio-1)
```

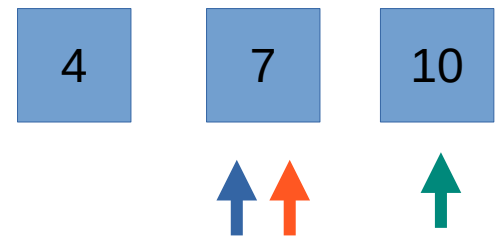
Ricerca Binaria - Esecuzione

N=7



STEP 1

$4 < 7$: cerchiamo a destra



STEP 2

$7 = 7$: trovato !

Mergesort Part 1

```
def merge_sort(lista):  
    if len(lista) <= 1:  
        return lista  
  
    metà = len(lista) // 2  
  
    sinistra = merge_sort(lista[:metà])  
    destra = merge_sort(lista[metà:])  
  
    return merge(sinistra, destra)
```

Mergesort Part 2

```
def merge(sinistra, destra):
    risultato = []
    i = j = 0

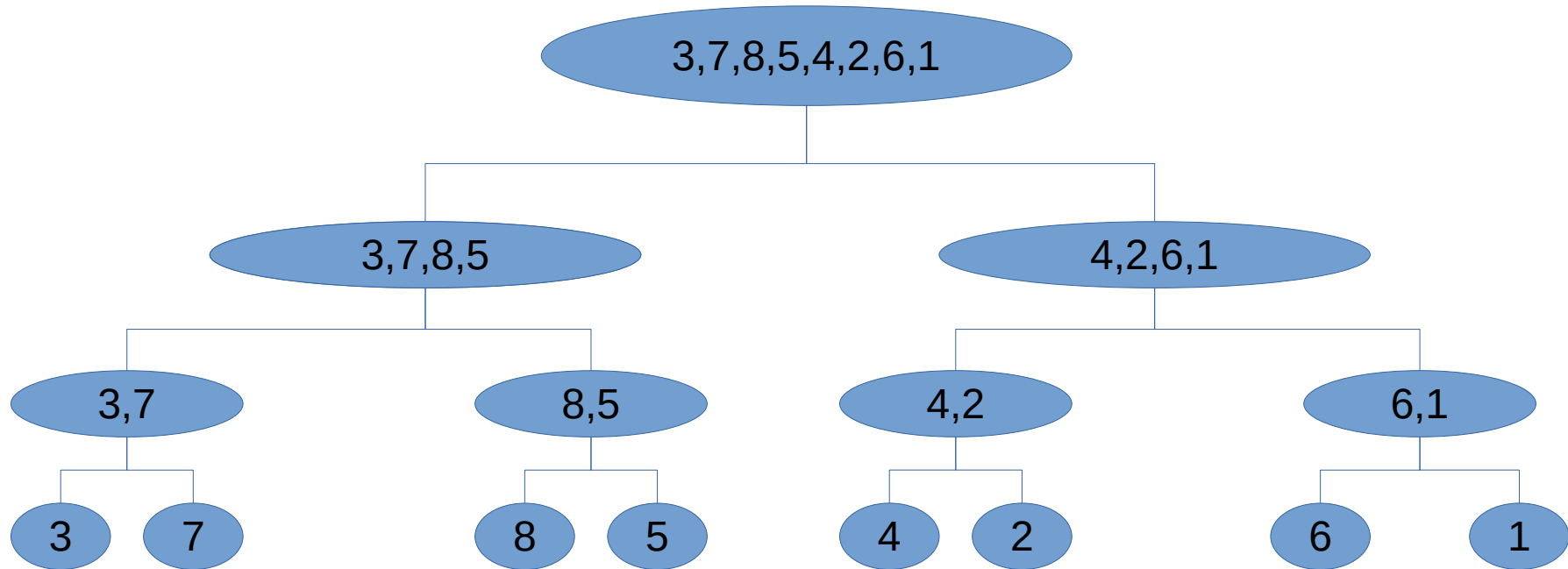
    # Two Finger algorithm: unisce due array ordinati
    while i < len(sinistra) and j < len(destra):
        if sinistra[i] < destra[j]:
            risultato.append(sinistra[i])
            i += 1
        else:
            risultato.append(destra[j])
            j += 1

    # Aggiunge gli elementi rimanenti
    risultato.extend(sinistra[i:])
    risultato.extend(destra[j:])
    return risultato
```

Mergesort - Uso

```
# Esempio di utilizzo  
Numeri = [3,8,7,5,4,2,6,1]  
print(merge_sort(numeri)) # Output: [1,2,3,4,5,6,7,8]
```

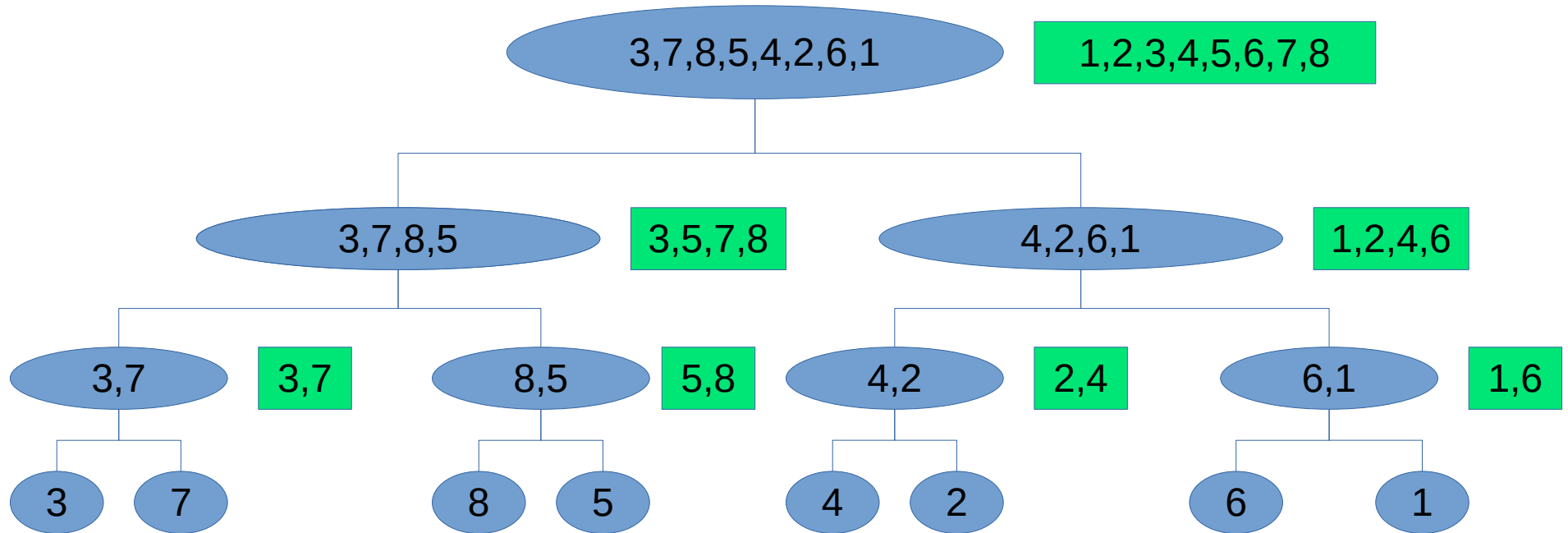
Mergesort – Esempio – Fase 1



COMPLESSITA: $2^{\text{steps}} = N = \text{len}(\text{lista})$

$\Rightarrow \text{steps} = \log_2 N \Rightarrow O(\log_2 N)$

Mergesort – Esempio – Fase 2



Mergesort – Esempio – Merge

Questa funzione Merge(), intermedia, che avviene nella funzione Mergesort() prima di restituire il valore, mette i valori in un nuovo array in modo crescente. Gli array di input devono essere già ordinati crescentemente.

3578 1246 → 1

3578 246 → 12

3578 46 → 123

578 46 → 1234

578 6 → 12345

78 6 → 123456

78 → 1234567

8 → 12345678