

# [FIM] FONDAMENTI DI INFORMATICA per medicina e chirurgia high tech

L05: Recursion

Dott. Giorgio De Magistris

*demagistris@diag.uniroma1.it*

CORSO DI LAUREA IN MEDICINA E CHIRURGIA HIGH TECH



SAPIENZA  
UNIVERSITÀ DI ROMA

I3S

FACOLTÀ DI INGEGNERIA DELL'INFORMAZIONE, INFORMATICA E STATISTICA

DIAG

DIPARTIMENTO DI INGEGNERIA INFORMATICA, AUTOMATICA E GESTIONALE

TUTTI I DIRITTI RELATIVI AL PRESENTE MATERIALE DIDATTICO ED AL SUO CONTENUTO SONO RISERVATI A SAPIENZA E AI SUOI AUTORI (O DOCENTI CHE LO HANNO PRODOTTO). È CONSENTITO L'USO PERSONALE DELLO STESSO DA PARTE DELLO STUDENTE A FINI DI STUDIO. NE È VIETATA NEL MODO PIÙ ASSOLUTO LA DIFFUSIONE, DUPLICAZIONE, CESSIONE, TRASMISSIONE, DISTRIBUZIONE A TERZI O AL PUBBLICO PENA LE SANZIONI APPLICABILI PER LEGGE

# Divide and Conquer

- Algorithmic approach to solve problems that can be subdivided into simpler problems similar to the original problem. It is based on the following steps:
  - Divide: break up the problem into smaller subproblems that are instances of the same problem but with a smaller input
  - Conquer: solve the subproblems by splitting them recursively until the size is small enough to compute the solution directly
  - Combine: combine the solution of the subproblems to find the solution of the original problem



# Example: linear search

- Assume you want to find an element in a list
- If the list is not sorted, I have to scan the list until I find the desired element
- The cost of the worst case scenario is  $\Theta(n)$

```
def linear_search(l, elem):  
    for e in l:  
        if e == elem:  
            return True  
    return False
```

**Time**

$n$

$n$

1

1

**Complexity**

$\Theta(n)$

# Example: Binary Search

- If the list is ordered I can do better than  $\Theta(n)$  using a Divide and Conquer algorithm: the binary search
- Divide: take the element in the middle of the list, if it is equal to the desired element return the element, if it is greater search in the sublist at the left of the middle element, otherwise search to the right
- Conquer: continue splitting until either the element is found or until I've a list with a single element
- Combine: If I found the element while splitting the list then return True. If I have a list with one element, if the element is the one I was looking for return true, else return false

# Example: Binary Search

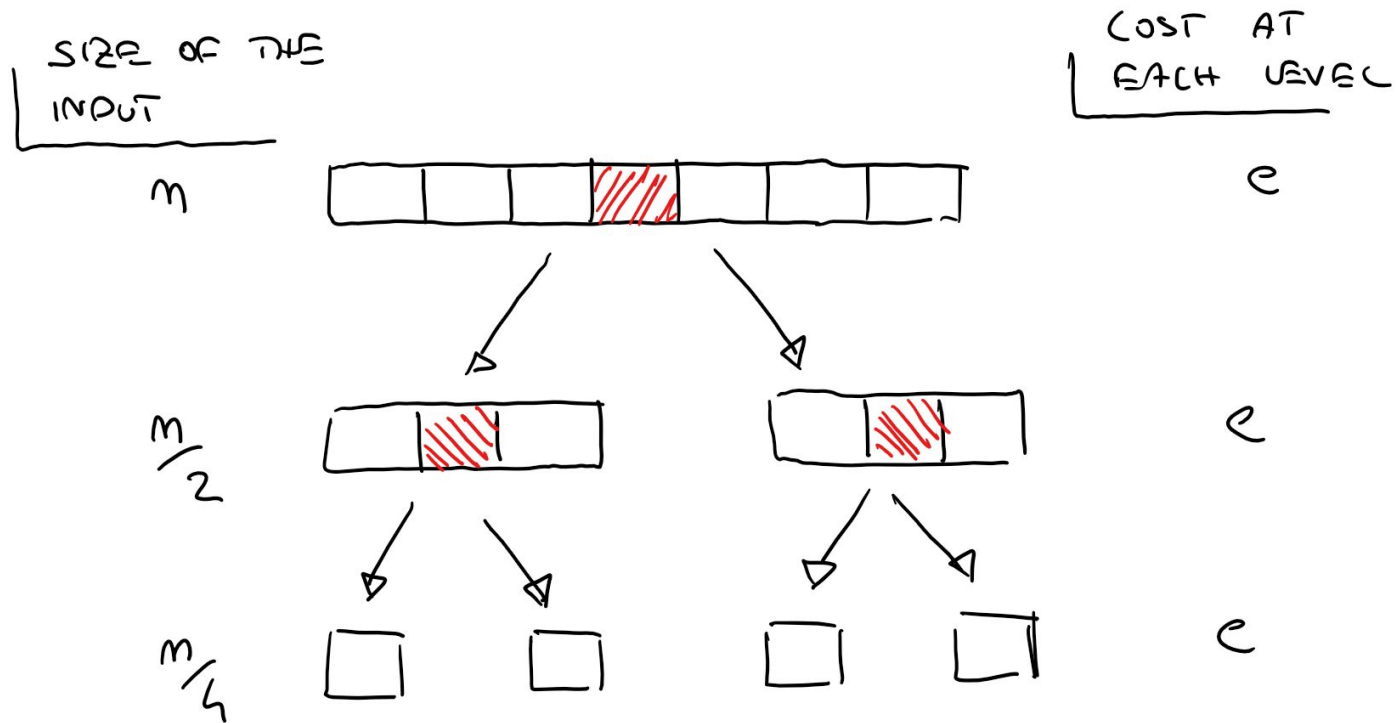
- This is the python implementation, but what is the cost?

```
def binary_search(l, elem):  
    # base case  
    if len(l) == 0:  
        return False  
    m = len(l) // 2  
    if elem == l[m]:  
        return True  
    # inductive step  
    elif elem > l[m]:  
        return binary_search(l[m+1:], elem)  
    else:  
        return binary_search(l[0:m], elem)
```

# Analysis

- There are different ways for analyzing a recursive algorithm
- The most straightforward is to draw the **recursion tree**
  - each node represents the cost of a single subproblem
  - we sum the costs within each level of the tree to obtain a set of per-level costs
  - then we sum all the per-level costs to determine the total cost of all levels of recursion

# Recursion-Tree



$$\text{TOTAL COST} = c \cdot (\# \text{ OF LEVELS}) = c \cdot \log_2 m$$

$$\Rightarrow \text{ASYMPTOTIC COST} = \Theta(\log m)$$

---

## **Slides distribuite con Licenza Creative Commons (CC BY-NC-ND 4.0) Attribuzione - Non commerciale - Non opere derivate 4.0 Internazionale**

### **PUOI CONDIVIDERLE ALLE SEGUENTI CONDIZIONI**

(riprodurre, distribuire, comunicare o esporre in pubblico, rappresentare, eseguire e recitare questo materiale con qualsiasi mezzo e formato)

#### **Attribuzione\***

Devi riconoscere una menzione di paternità adeguata, fornire un link alla licenza e indicare se sono state effettuate delle modifiche. Puoi fare ciò in qualsiasi maniera ragionevole possibile, ma non con modalità tali da suggerire che il licenziante avalli te o il tuo utilizzo del materiale.

#### **Non Commerciale**

Non puoi utilizzare il materiale per scopi commerciali.

#### **Non opere derivate**

Se remixi, trasformi il materiale o ti basi su di esso, non puoi distribuire il materiale così modificato.

#### **Divieto di restrizioni aggiuntive**

Non puoi applicare termini legali o misure tecnologiche che impongano ad altri soggetti dei vincoli giuridici a questa licenza