

TECNOLOGIE INFORMATICHE MULTIMEDIALI

««Gli errori sono necessari, utili come il pane e spesso anche belli: per esempio, la torre di Pisa.»»

(Gianni Rodari, Il libro degli errori)

ALGORITMI DI RICERCA DI INFORMAZIONI

Algoritmi di visitaazione degli alberi

perché

La formalizzazione di un problema è spesso rappresentabile con una struttura ad **ALBERO**

Tipologia

NON INFORMATI: nessuna conoscenza specifica e generalità (**brute force**)

Tipologia

EURISTICI: conoscenze specifiche

Tipologia

GIOCHI: ricerca con agente ostile («avversario»)

- **RICERCA IN SALITA (Hill Climbing)**
- **RICERCA BEST FIRST**
- **RICERCA GOLOSA (Greedy Search)**
- **RICERCA BRANCH AND BOUND**
- **RICERCA A E A***

ALGORITMI DI RICERCA DI INFORMAZIONI

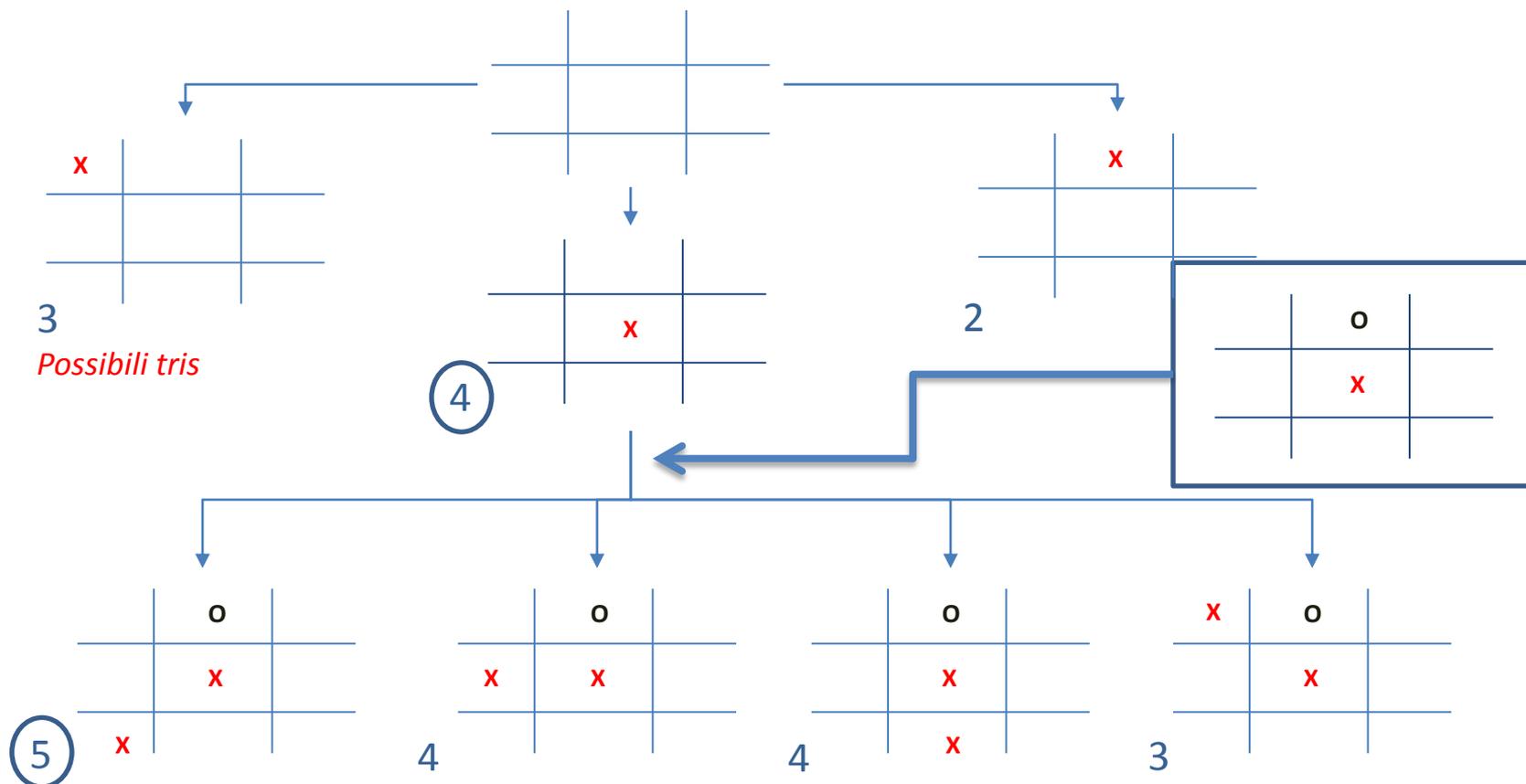
Algoritmi di visitaione degli alberi

perché

La formalizzazione di un problema è spesso rappresentabile con una struttura ad **ALBERO**

Algoritmi Euristici: (*valutazione euristica*)

Esempio nel il gioco del TIC TAC TOE (tris) si valuta il numero potenziale di tris



ALGORITMI DI RICERCA DI INFORMAZIONI

Algoritmi di visita degli alberi

perché

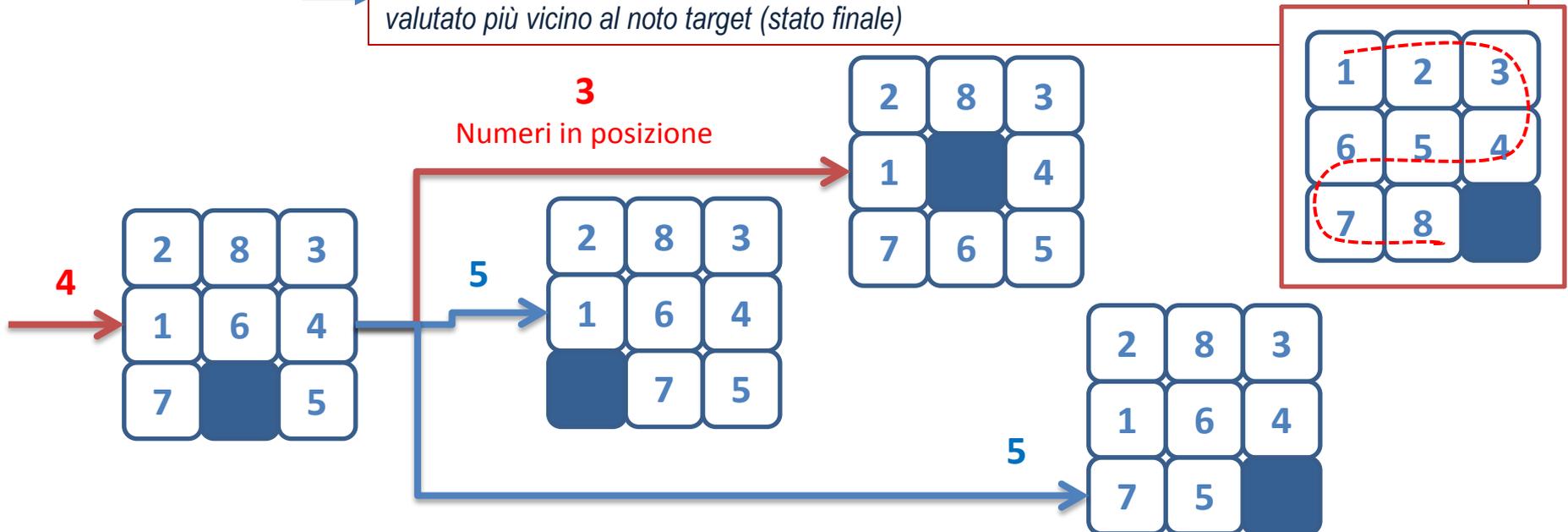
La formalizzazione di un problema è spesso rappresentabile con una struttura ad **ALBERO**

Algoritmi Euristicici: (*valutazione euristica*)

Usa **INFORMAZIONI CARATTERISTICHE** del problema per decidere quale nodi espandere

All'insieme delle configurazioni (spazio degli stati) associa una **FUNZIONE EURISTICA** che valuta gli stati e gli assegna un valore (valorizza ogni nodo-stato)

Applica una **STRATEGIA EURISTICA**: sceglie il nodo più «interessante», con «migliori prospettive» e valutato più vicino al noto target (stato finale)

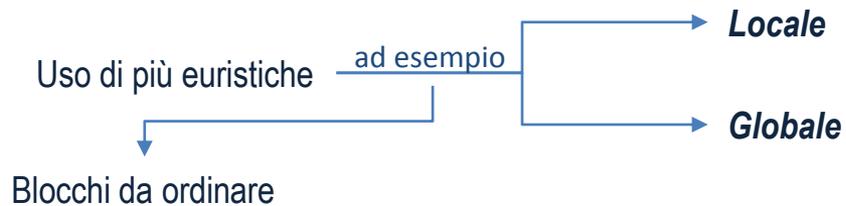


ALGORITMI DI RICERCA DI INFORMAZIONI

ALGORITMI EURISTICI

- **RICERCA IN SALITA (HILL CLIMBING)**
- *RICERCA BEST-FIRST*
- *RICERCA GOLOSA (GREEDY SEARCH)*
- *RICERCA BRANCH AND BOUND*
- *RICERCA A E A**

Selezione dello stato successivo in modo dipendente dal criterio:
«migliora lo stato secondo l'euristica che guida il procedimento»



1 – stato finale



2 – stato iniziale



3 – primo stato



4 – stato possibile da stato 3



5 – stato possibile da stato 3

EURISTICA LOCALE : ogni stato ha +1 per un blocco in posizione corretta e -1 per ogni blocco in posizione errata

STATO 1, +4
STATO 2, 0
STATO 3, +2 *stop*
STATO 4, 0
STATO 5, 0

EURISTICA GLOBALE: ogni stato ha +n per ogni blocco sopra n-1 blocchi corretti e -n per ogni blocco sopra n-1 errati

STATO 1, +10
STATO 2, -10
STATO 3, -5
STATO 4, -4
STATO 5, -3

ALGORITMI DI RICERCA DI INFORMAZIONI

- fallibili
- difficili
 - ✓ da individuare
 - ✓ da verificare sperimentalmente

ALGORITMI EURISTICI

- **RICERCA IN SALITA (HILL CLIMBING)**
- *RICERCA BEST-FIRST*
- *RICERCA GOLOSA (GREEDY SEARCH)*
- *RICERCA BRANCH AND BOUND*
- *RICERCA A E A**

Selezione dello stato successivo in modo dipendente dal criterio:
«migliora lo stato secondo l'euristica che guida il procedimento»

Problemi / limiti

MASSIMO LOCALE

Lo stato trovato è migliore degli stati vicini ma non di quelli lontani

SOLUZIONE

Backtracking (passo indietro) e nuova direzione

PIANORO

Tutti gli stati prossimi tra loro hanno pari valore

SOLUZIONE (simulated annealing)

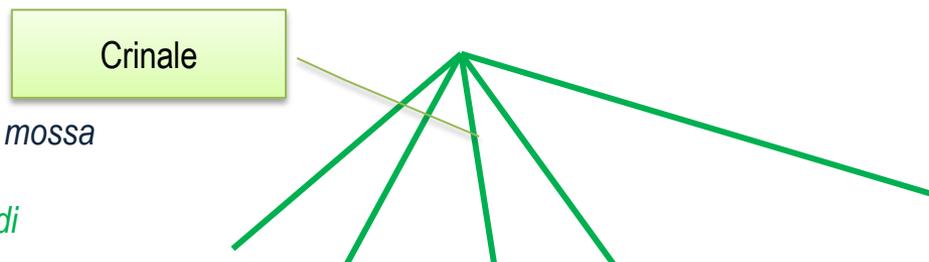
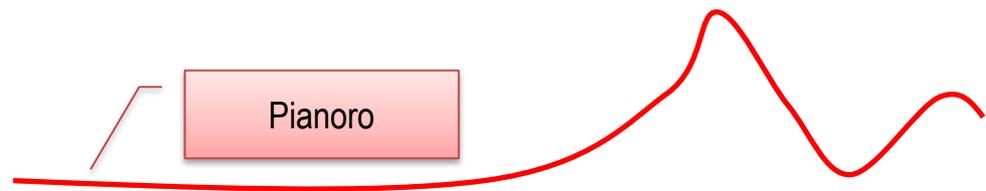
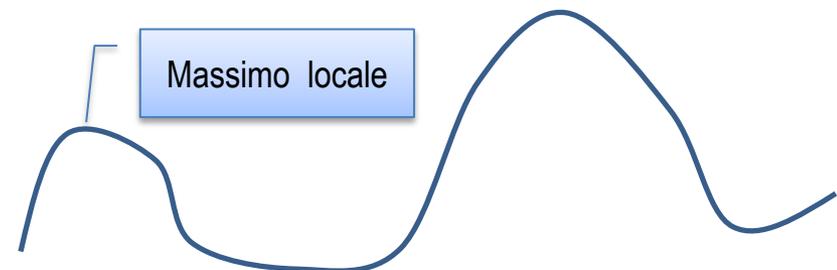
saltare in qualche direzione verso una nuova zona

CRINALE

Ci sono stati prossimi migliore ma lontano più di una mossa

SOLUZIONE

fare due o più mosse insieme (AGGIRAMENTO) prima di valutare i prossimi stati (MACROREGOLE)

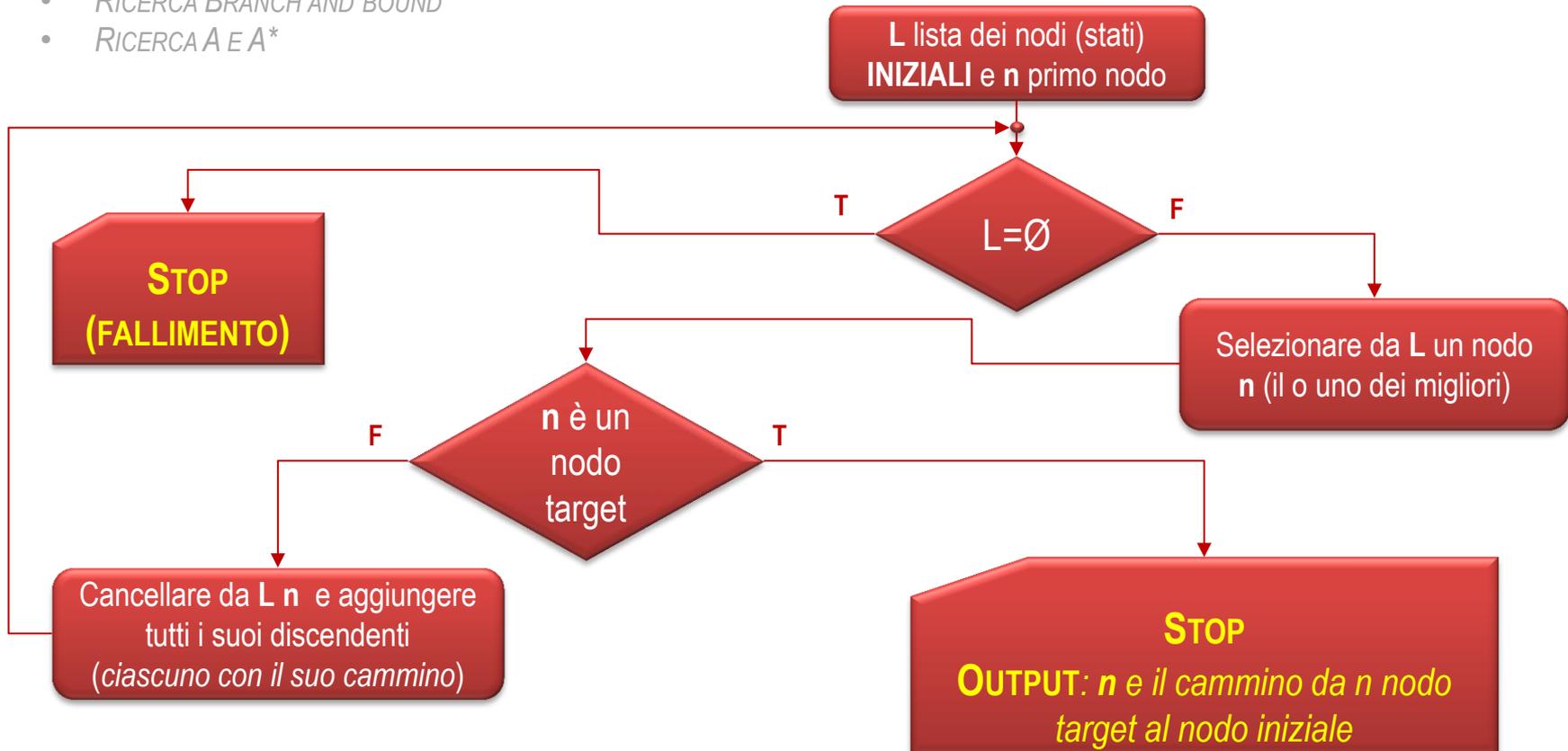


ALGORITMI DI RICERCA DI INFORMAZIONI

ALGORITMI EURISTICI

- RICERCA IN SALITA (HILL CLIMBING)
- **RICERCA BEST-FIRST**
- RICERCA GOLOSA (GREEDY SEARCH)
- RICERCA BRANCH AND BOUND
- RICERCA A E A*

Strategia «cerca prima il migliore» adatto a problemi di intelligenza artificiale. La ricerca best-first si basa su una funzione $f(n)$ che seleziona ad ogni passo il nodo successivo, quello con $f(n)$ più basso.



ALGORITMI DI RICERCA DI INFORMAZIONI

ALGORITMI EURISTICI

- RICERCA IN SALITA (HILL CLIMBING)
- RICERCA BEST-FIRST
- **RICERCA GOLOSA (GREEDY SEARCH)**
- RICERCA BRANCH AND BOUND
- RICERCA A E A*

Esempio

Il problema «**dai il minor numero di monete di resto utilizzando monete da 100, 10, 1 cent**» è un problema risolvibile tramite un algoritmo di tipo GREEDY SEARCH: ad ogni passo viene controllato il resto ancora da dare e si aggiunge la moneta con valore maggiore possibile. Quindi per dare un resto di 112 cent la macchina farà cadere in sequenza una moneta da 100, poi 10, poi 1, e infine ancora 1 cent (4 monete)...
SE CI FOSSERO LE MONETE DA 112 CENT?

Un algoritmo **greedy** ha come obiettivo quello di ottenere una soluzione ottima da un punto di vista globale attraverso la scelta della soluzione più golosa (aggressiva o avida) ad ogni passo locale.

Minimizzazione della distanza dall'obiettivo ←

Definizione di una funzione euristica h che valuta per ogni stato la distanza dall'obiettivo [se n è l'obiettivo $h(n) = 0$]

Simile alla ricerca «Ricerca in Salita» (Hill-Climbing) ma vero e proprio albero: si espandono sempre i nodi valutati più vicini all'obiettivo

Simile alla Ricerca in Profondità e non ottimale, ma se la funzione euristica è buona può essere molto veloce

Trovare una buona funzione H (euristica) è difficile

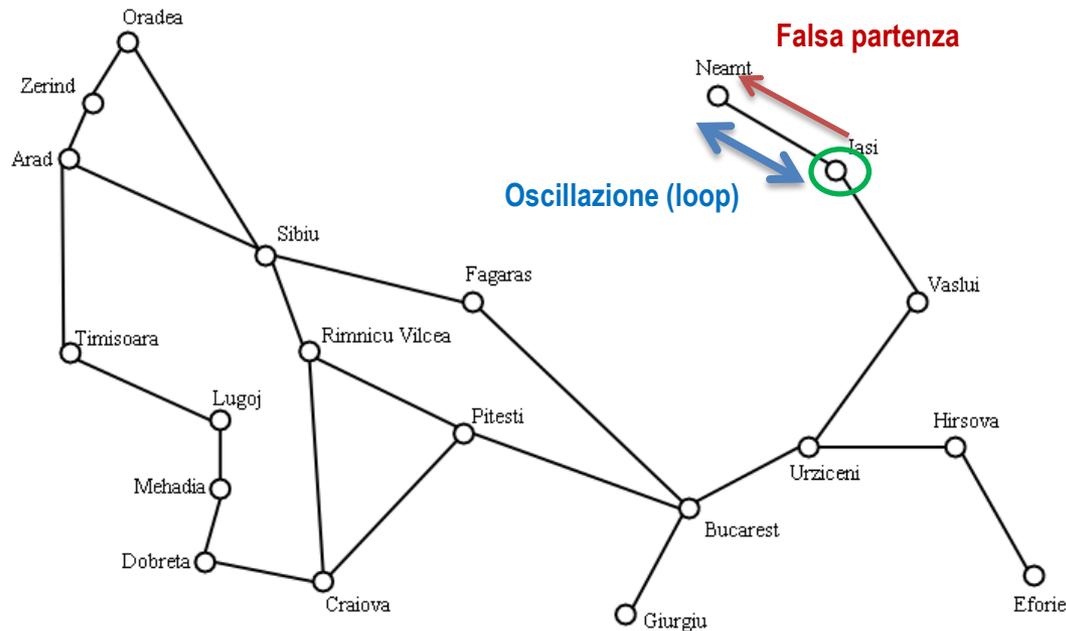
ALGORITMI DI RICERCA DI INFORMAZIONI

ALGORITMI EURISTICI

- RICERCA IN SALITA (HILL CLIMBING)
- RICERCA BEST-FIRST
- **RICERCA GOLOSA (GREEDY SEARCH)**
- RICERCA BRANCH AND BOUND
- RICERCA A E A*

Un algoritmo **greedy** ha come obiettivo quello di ottenere una soluzione ottima da un punto di vista globale attraverso la scelta della soluzione più golosa (aggressiva o avida) ad ogni passo locale.

L'inserimento di un nuovo nodo può rendere la soluzione non ottimale



$h(n)$ = distanza in linea d'aria
(funzione euristica)

ALGORITMI DI RICERCA DI INFORMAZIONI

ALGORITMI EURISTICI

- RICERCA IN SALITA (HILL CLIMBING)
- RICERCA BEST-FIRST
- RICERCA GOLOSA (GREEDY SEARCH)
- **RICERCA BRANCH AND BOUND**
- RICERCA A E A*

Un algoritmo che sfrutta una tecnica applicabile a problemi con uno **SPAZIO FINITO DI SOLUZIONI** e che si fonda sulla scomposizione del problema in sotto-problemi più semplici da risolvere.

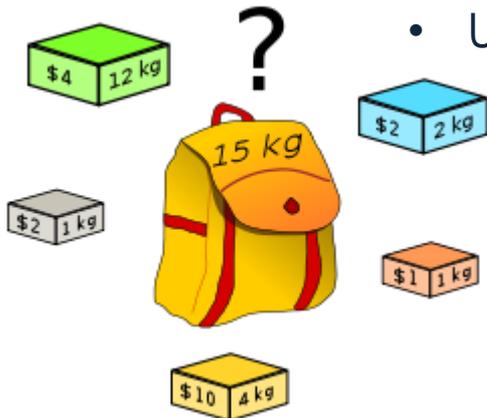
A. H. Land e A. G. Doig (1960)

garantiscono

Algoritmi di ENUMERAZIONE IMPLICITA, si comportano come un algoritmo di enumerazione: «provano» tutte le soluzioni possibili fino a trovare quella ottima, o quella corretta, e ne scartano alcune dimostrando a priori la loro non **OTTIMALITÀ**.

sono

- Trova sicuramente il cammino di costo minimale
- Utile quando esistono più cammini da nodo iniziale a nodo finale



IL PROBLEMA DELLO ZAINO (*Knapsack problem*)

problema di ottimizzazione combinatoria

«Sia dato uno zaino che possa sopportare un determinato peso. Siano dati inoltre N oggetti, ognuno dei quali caratterizzato da un peso e un valore. Il problema si propone di scegliere quali di questi oggetti mettere nello zaino per ottenere il maggiore valore senza eccedere nel peso sostenibile dallo zaino stesso.»

ALGORITMI DI RICERCA DI INFORMAZIONI

ALGORITMI EURISTICI

- RICERCA IN SALITA (HILL CLIMBING)
- RICERCA BEST-FIRST
- RICERCA GOLOSA (GREEDY SEARCH)
- RICERCA BRANCH AND BOUND
- **RICERCA A E A***

Un algoritmo best-first con una funzione di valutazione totale:

$$f(n) = g(n) + h(n)$$

$f(n)$: costo totale del cammino dal
NODO INIZIALE al NODO
FINALE passando per il nodo n

$g(n)$: misura della lunghezza effettiva del
cammino dal NODO INIZIALE a n

$h(n)$: stima euristica della distanza del nodo
 n dal NODO FINALE

ALGORITMI DI RICERCA DI INFORMAZIONI

$g(n)$: misura della lunghezza effettiva del cammino dal NODO INIZIALE a n

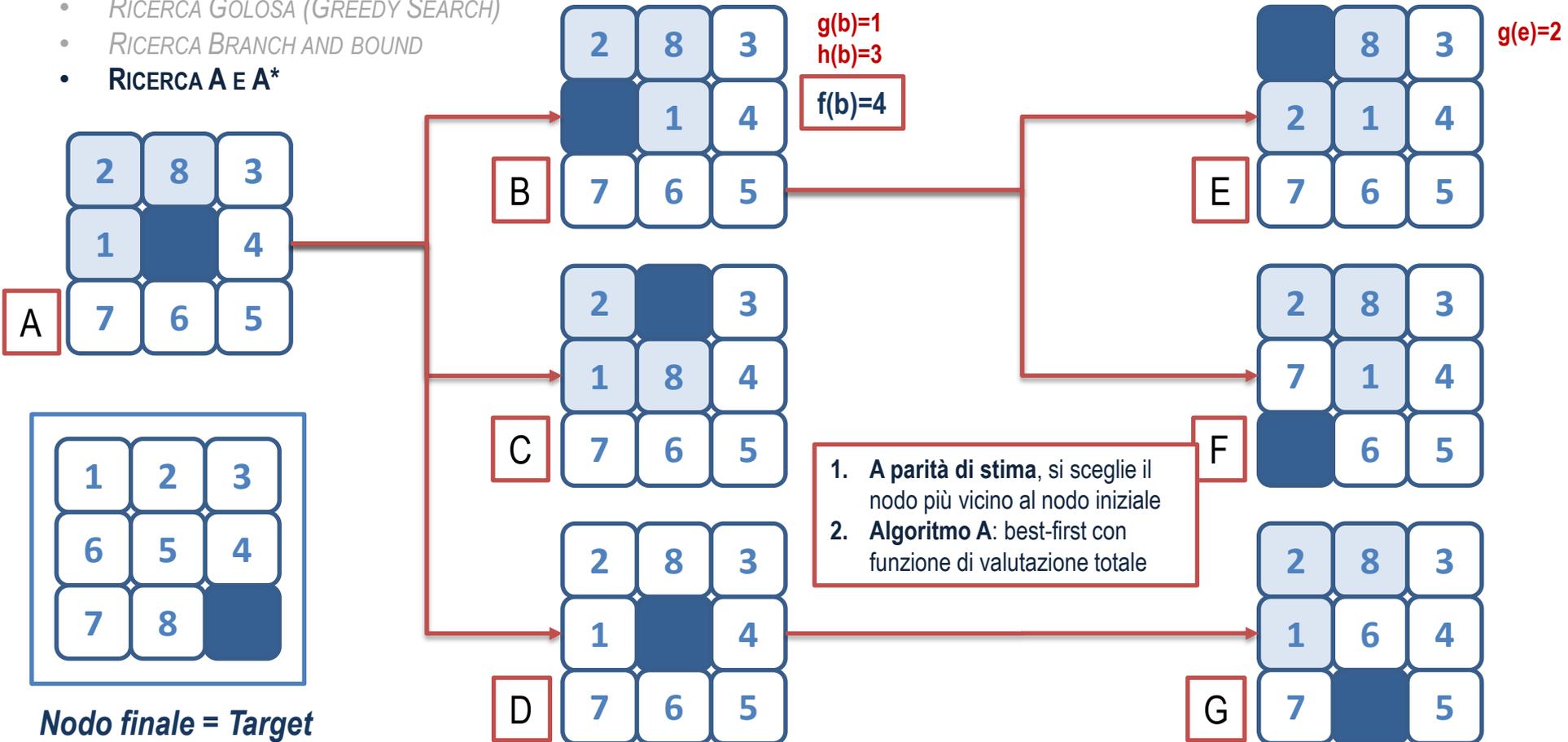
$h(n)$: stima euristica della distanza del nodo n dal NODO FINALE

Un algoritmo best-first con una funzione di valutazione totale, **ricerca A**:

$$f(n) = g(n) + h(n)$$

ALGORITMI EURISTICI

- RICERCA IN SALITA (HILL CLIMBING)
- RICERCA BEST-FIRST
- RICERCA GOLOSA (GREEDY SEARCH)
- RICERCA BRANCH AND BOUND
- RICERCA A E A*



ALGORITMI DI RICERCA DI INFORMAZIONI

ALGORITMI EURISTICI

- RICERCA IN SALITA (HILL CLIMBING)
- RICERCA BEST-FIRST
- RICERCA GOLOSA (GREEDY SEARCH)
- RICERCA BRANCH AND BOUND
- RICERCA A E A*

A* è un algoritmo con FUNZIONE DI VALUTAZIONE TOTALE dove la funzione euristica h è **AMMISSIBILE**¹

$$f^*(n) = g^*(n) + h^*(n)$$

Peter Hart, Nils Nilsson, e
Bertram Raphael (1968)

A* è OTTIMAMENTE EFFICIENTE: non esiste nessun altro algoritmo ottimale che garantisce di espandere un minor numero di nodi di A*

$f^*(n)$: costo effettivo del
CAMMINO OTTIMALE dal NODO
INIZIALE al NODO FINALE
passando per il nodo n

$g^*(n)$: costo del cammino più corto della
dal NODO INIZIALE a n

$h^*(n)$: costo effettivo del cammino più
corto dal nodo n al NODO FINALE

NON NOTE f^*, g^*, h^*

approssimate da

f, g, h

con

$$g(n) \geq g^*(n)$$

per A*

è

$$h(n) \leq h^*(n)$$

- OTTIMALE
- COMPLETO

(1) Una funzione è **AMMISSIBILE** se non **sopravaluta** mai la distanza effettiva all'obiettivo

ALGORITMI DI RICERCA DI INFORMAZIONI

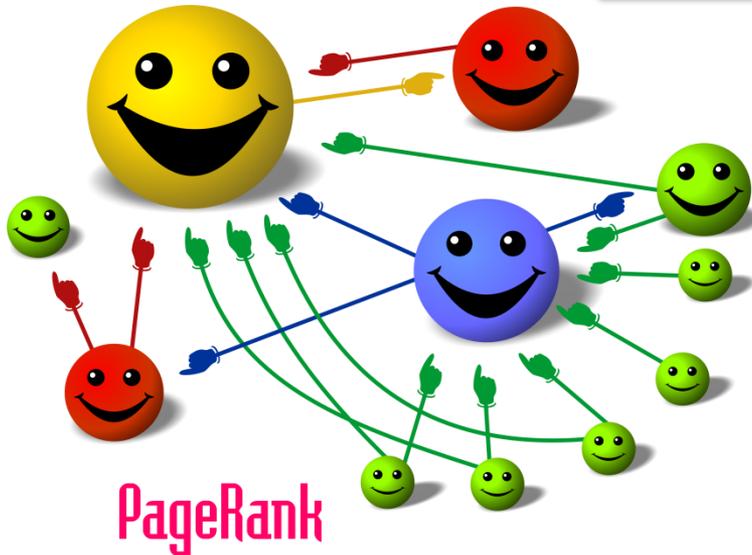
EFFICACIA DELLE EURISTICHE

	COSTO DELLA RICERCA		
p	ID (Iterative Deepening)	A*(h ₁)	A*(h ₂)
2	10	6	6
4	112	13	12
6	680	20	18
8	6384	39	25
10	47127	93	39
12	364404	227	73
14	3473941	539	113

ALGORITMI DI RICERCA DI INFORMAZIONI

ALGORITMO DI ANALISI

Assegnazione di un peso ad ogni elemento di collegamento ipertestuale di un insieme di documenti



PageRank è un marchio Google

Algoritmo brevettato dall'università di Stanford

$PR(E)$ = PageRank di E è il valore, peso numerico, associato ad un elemento E

- Una pagina è tanto più importante quanto più numerose sono le pagine che la puntano

Nomen omen...

Algoritmo ideato, tra gli altri da LAWRENCE EDWARD "LARRY" PAGE fondatore, con SERGEY BRIN, di Google.