TECNOLOGIE INFORMATICHE MULTIMEDIALI

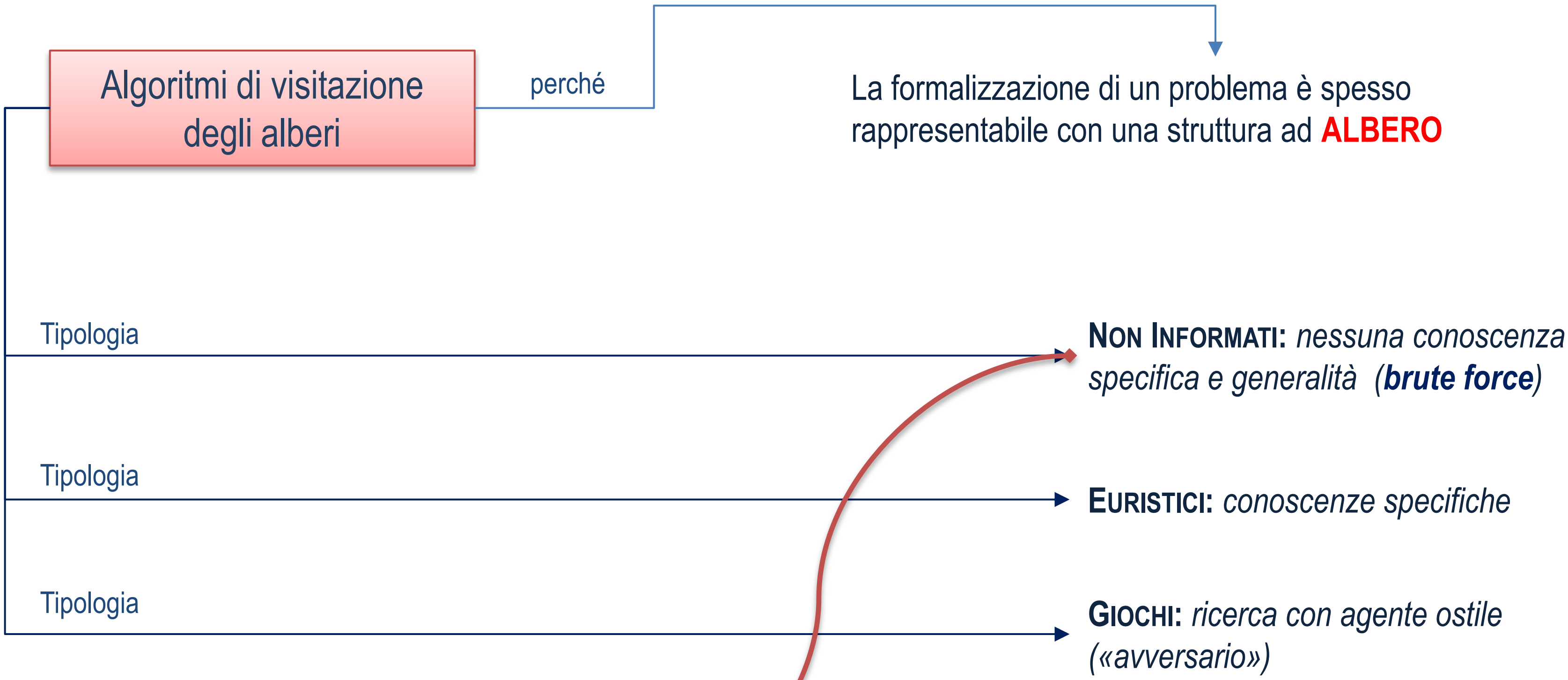
«Si troverà sempre una cosa nell'ultimo posto dove la si cerca.»

(Arthur Bloch, Legge di Boob, Il secondo libro di Murphy)

Università degli Studi di Ferrara
Dipartimento di Studi Umanistici
Corso di Laurea in «Scienze e Tecnologie della Comunicazione»
aa 2012-2013

Prof. Giorgio Poletti
giorgio.poletti@unife.it

Algoritmi di Ricerca di Informazioni



- **Ricerca in Profondità (depth-first)**
- **Ricerca in Larghezza (breadth-first)**
- **Iterative Deepening Algorithm (IDA)**

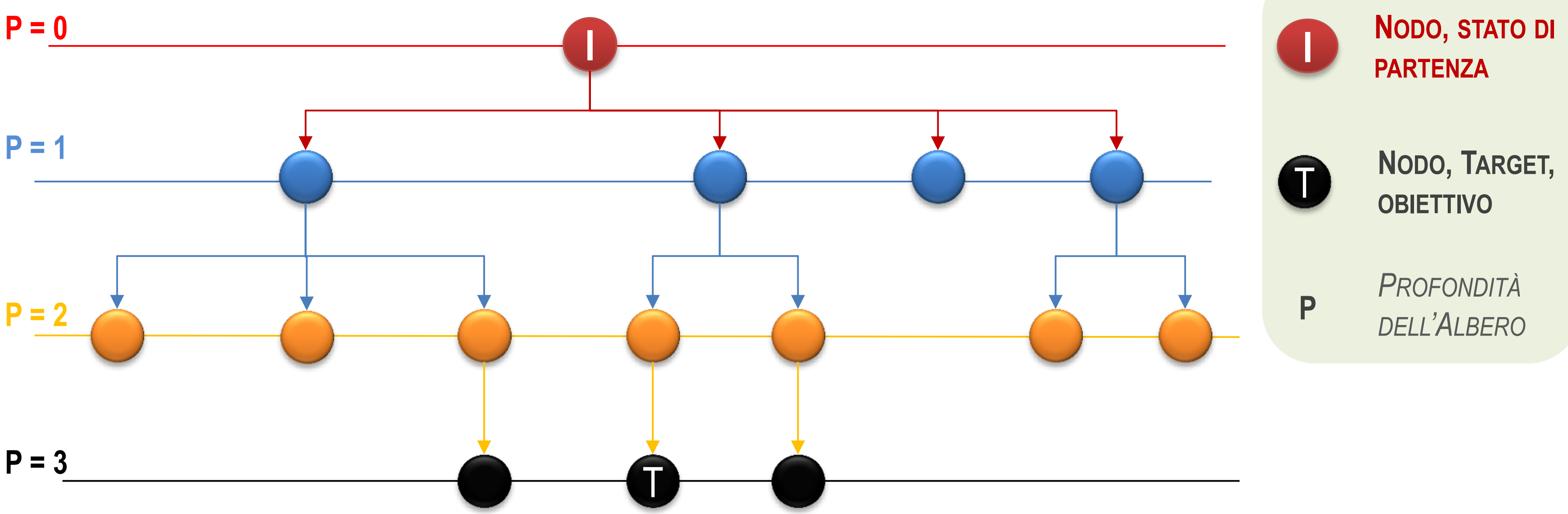
Algoritmi di Ricerca di Informazioni

Algoritmi di visitazione degli alberi

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

Obietivo → **Trovare** un percorso che unisca il nodo «stato di partenza» al nodo «obiettivo» (*valore cercato*)

Albero di Ricerca



Profondità 3

ALGORITMI DI RICERCA DI INFORMAZIONI

Algoritmi di visitazione degli alberi

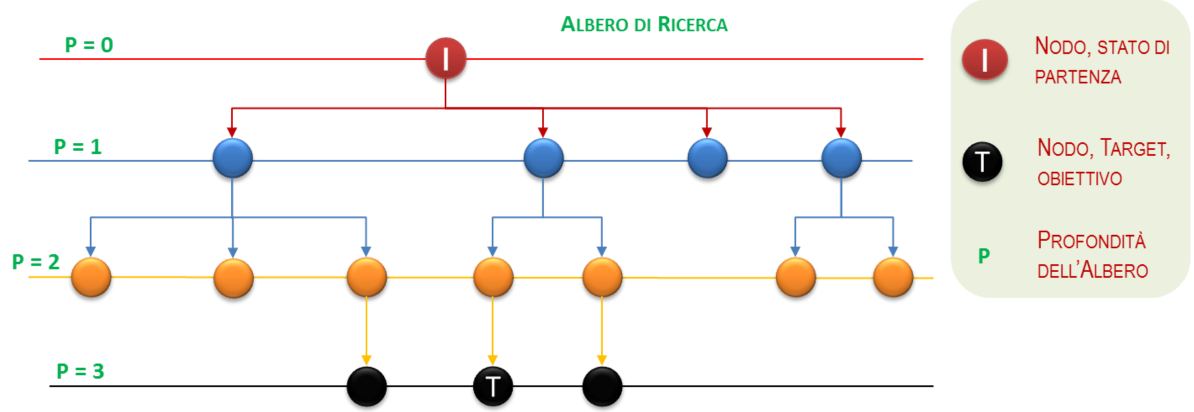
perché

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

formato da

NODI: *strutture dati*

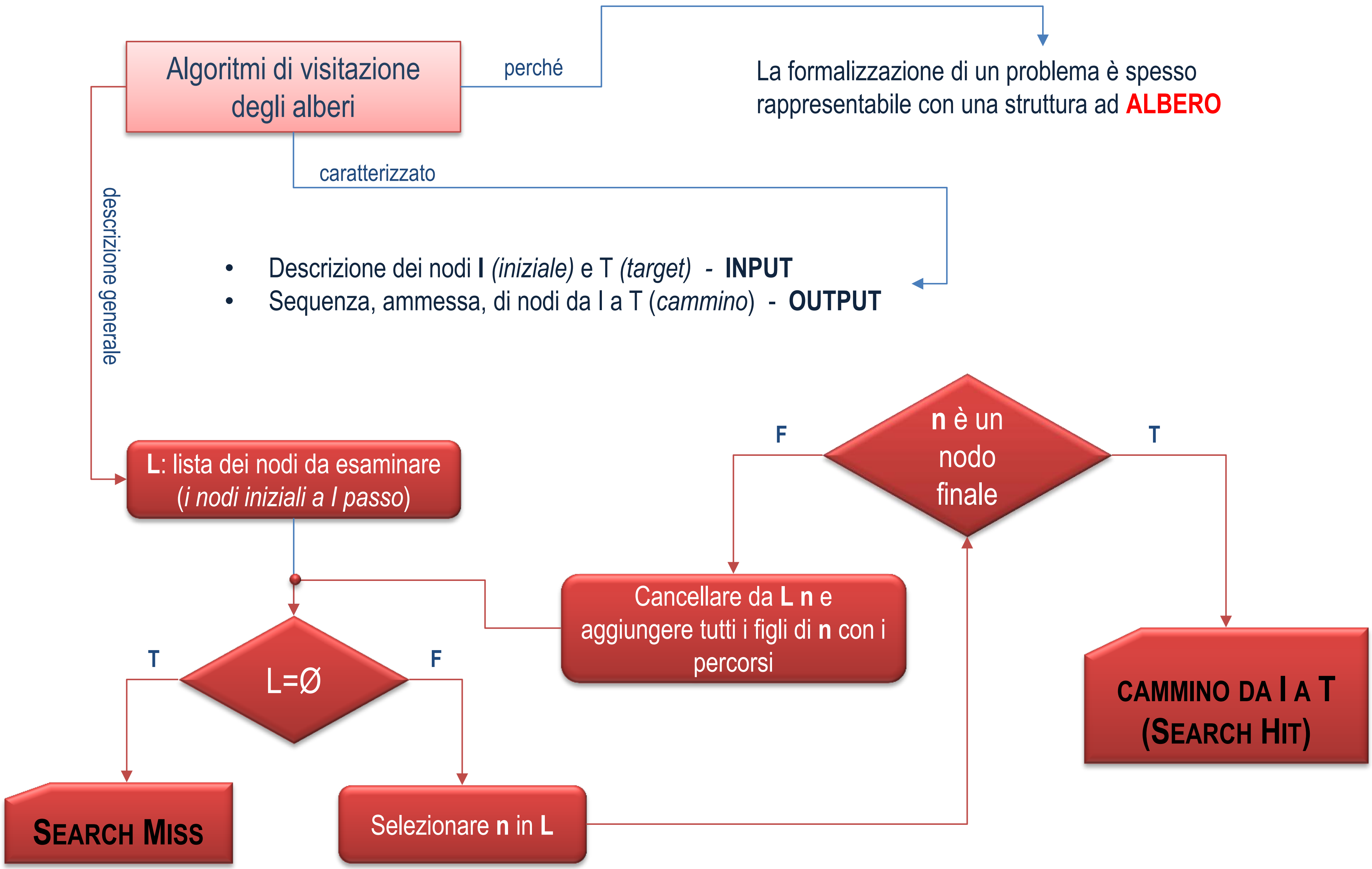
notazione *Stuart Russell* e *Peter Norvig*



NODI: *n-upla* ***n{s,nGen,o,p,c}***, con

- **s:** stato rappresentato dal nodo
- **nGen:** nodo padre (genitore)
- **o:** operatore che produce n
- **p:** profondità del nodo
- **c:** costo del cammino che unisce I (stato di partenza o iniziale a n)

Algoritmi di Ricerca di Informazioni

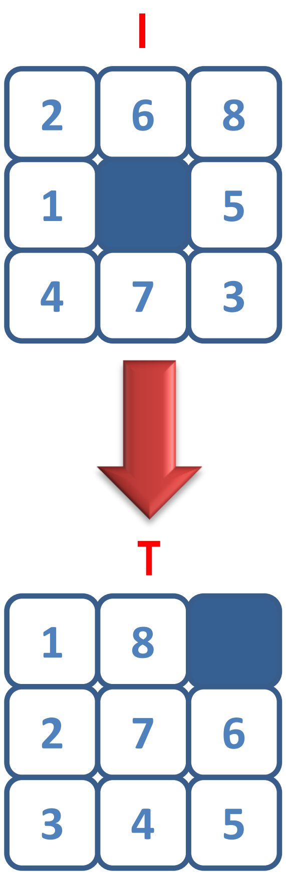


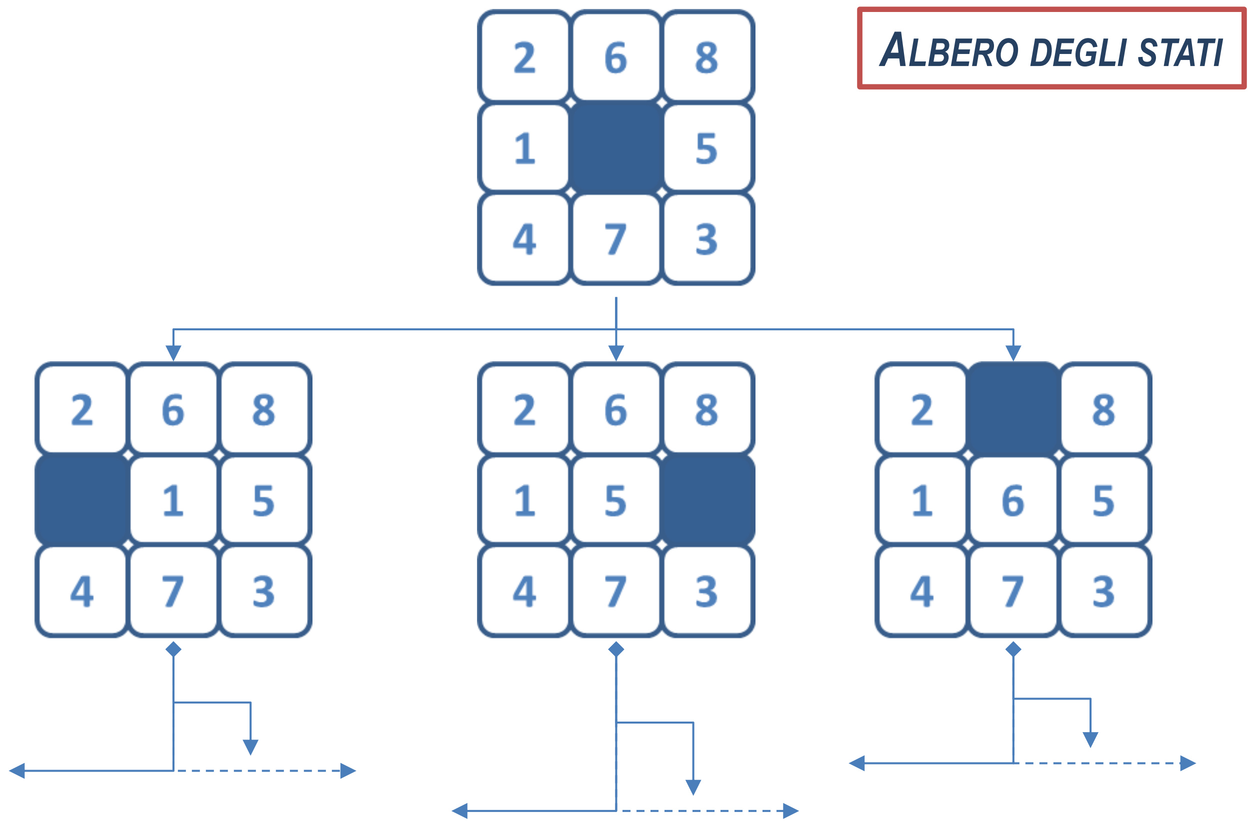
ALGORITMI DI RICERCA DI INFORMAZIONI

Algoritmi di visitazione degli alberi

perché

ALBERO DI RICERCA per avere una configurazione di tessere numerate a partire da una data



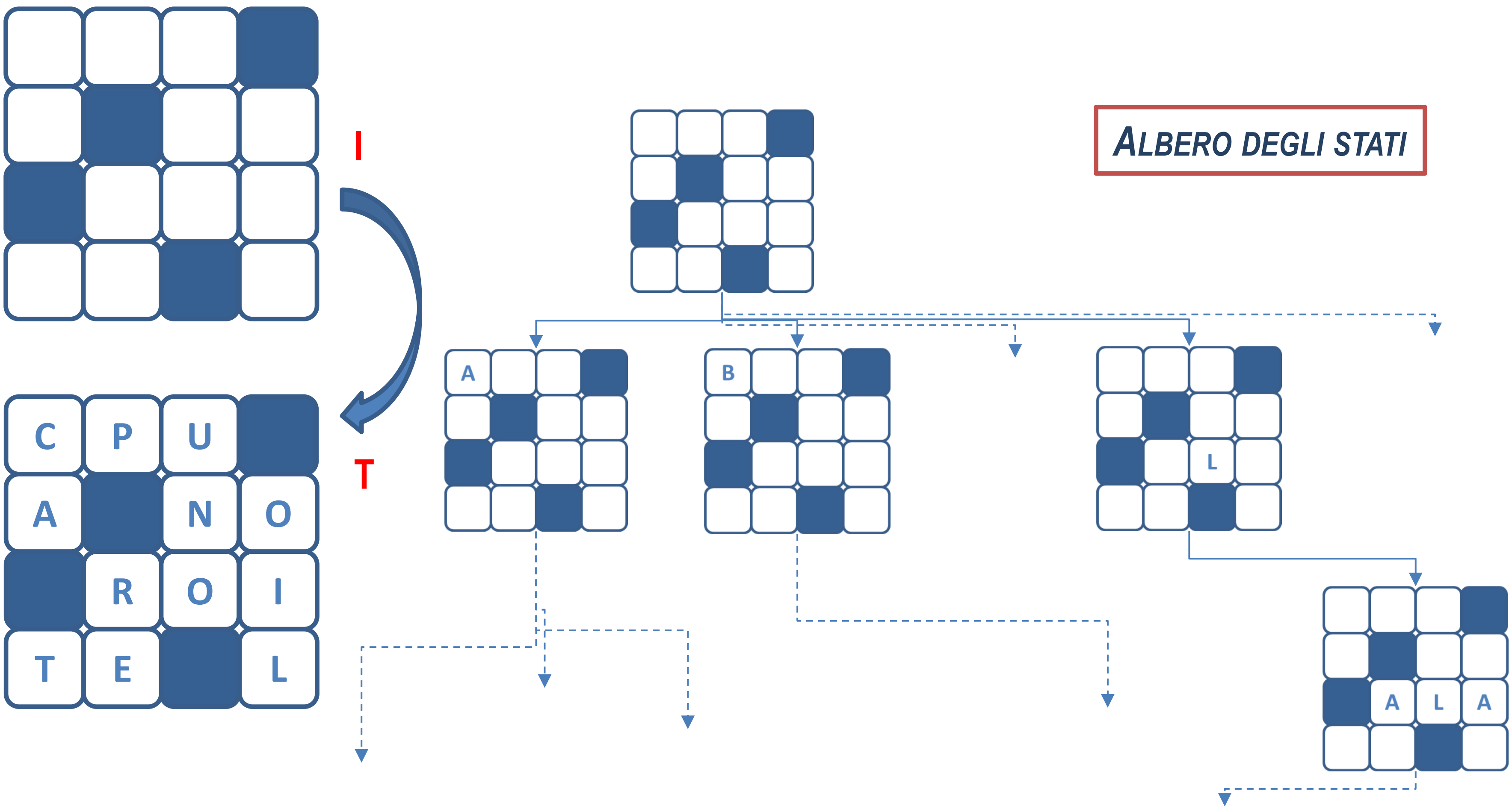


Algoritmi di Ricerca di Informazioni

Algoritmi di visitazione degli alberi

perché

Albero di ricerca per avere una configurazione di tessere numerate a partire da una data



ALGORITMI DI RICERCA DI INFORMAZIONI

Algoritmi di visitazione degli alberi (criteri valutativi)

COMPLESSITÀ

- **SPAZIALE**: *quantità di memoria necessaria*

- **TEMPORALE**: *quantità di tempo necessario*

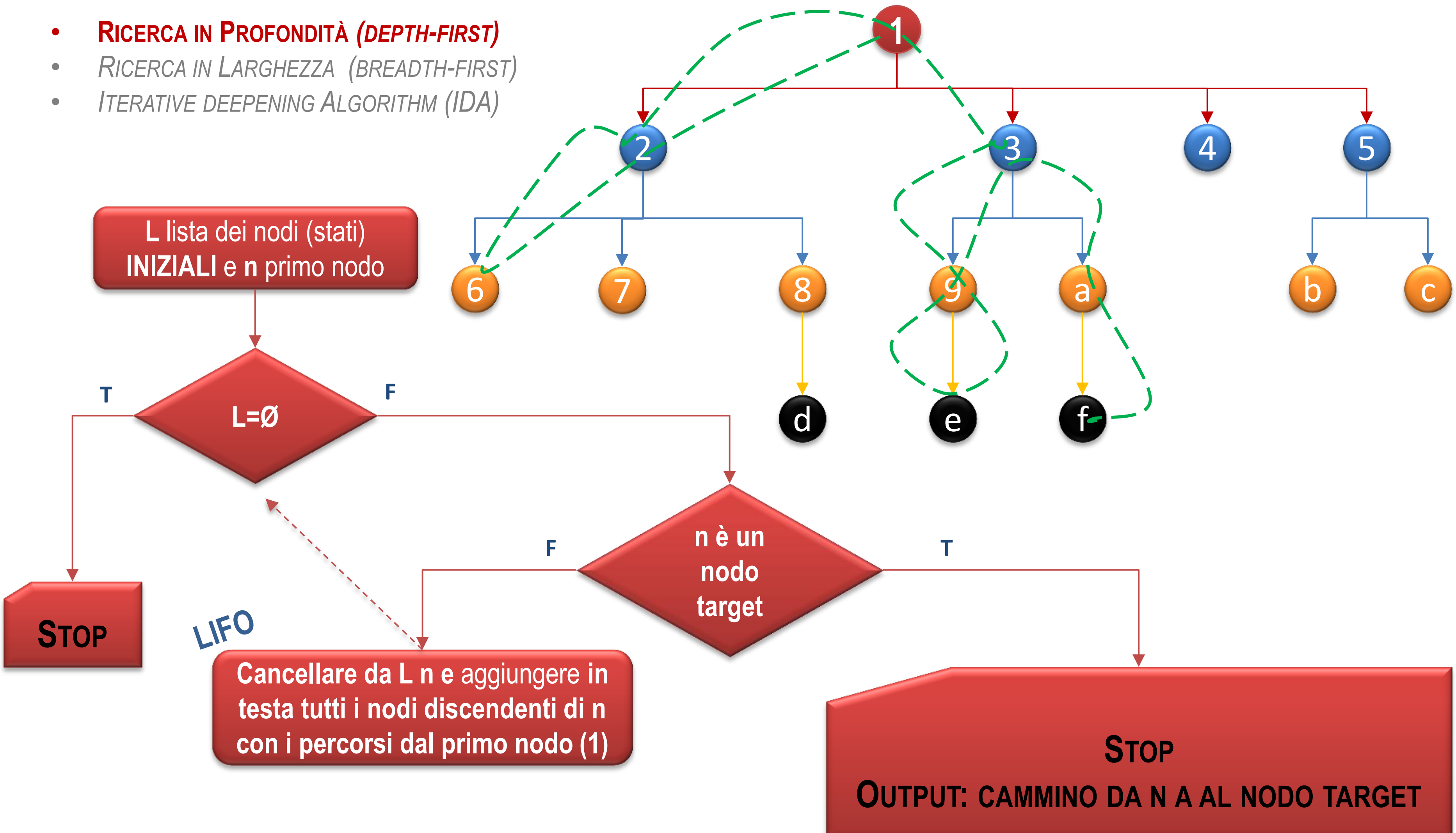
OTTIMALITÀ: *capacità dell'algoritmo di individuare la soluzione ottimale tra quelle trovate*

COMPLETEZZA: *capacità dell'algoritmo di individuare una soluzione se esiste*

Algoritmi di Ricerca di Informazioni

Algoritmi di visitazione degli alberi

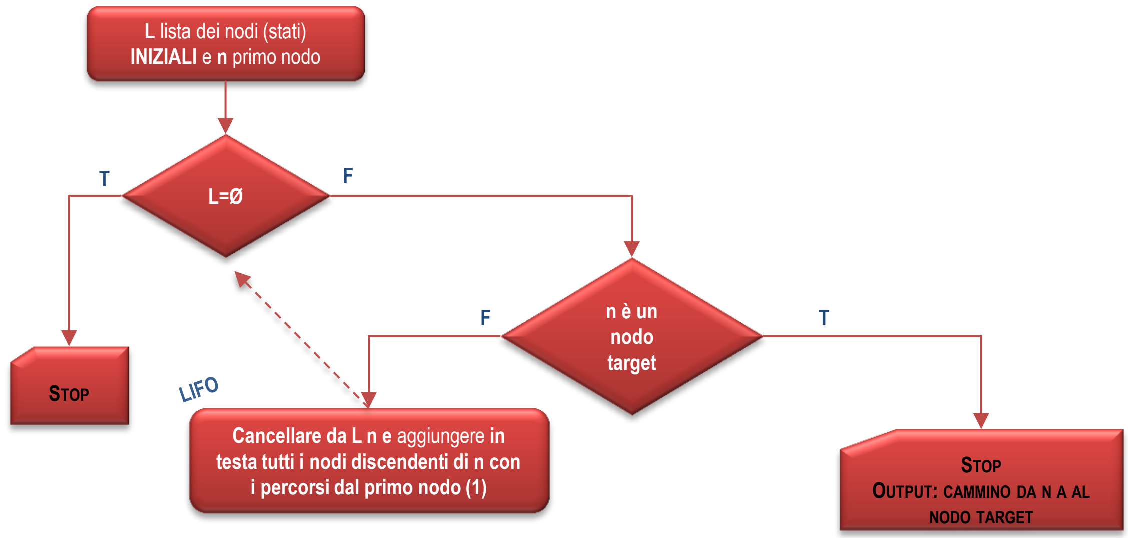
- **Ricerca in Profondità *(depth-first)***
- *Ricerca in Larghezza (breadth-first)*
- *Iterative deepening Algorithm (IDA)*



Algoritmi di Ricerca di Informazioni

Algoritmi di visitazione degli alberi

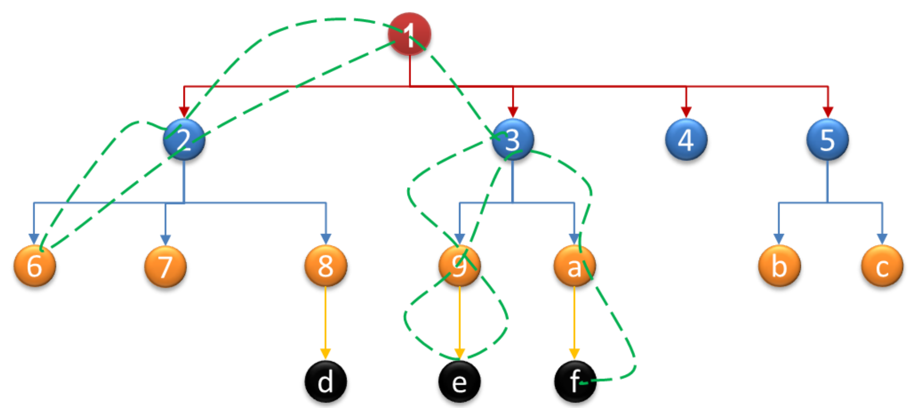
- **Ricerca in Profondità *(depth-first)***
- *Ricerca in Larghezza (breadth-first)*
- *Iterative deepening Algorithm (IDA)*



Complessità Spaziale
con **p** (*profondità*) e **b** (*bilanciamento*)

O(bp)[1]

Quantità massima di memoria ***p(b-1)+1***



Complessità Temporale
Caso Migliore, soluzione estremità sinistra dell'albero (**nodo 6**)
Caso Peggiore, soluzione estremità sinistra dell'albero (**nodo c**)

nodi da esaminare

Caso Migliore *$p+1$*
Caso Peggiore *$1+\ldots+b^p = (b^{p+1}-1)\,/\,/(b-1)$*

con p elevato ordine di grandezza

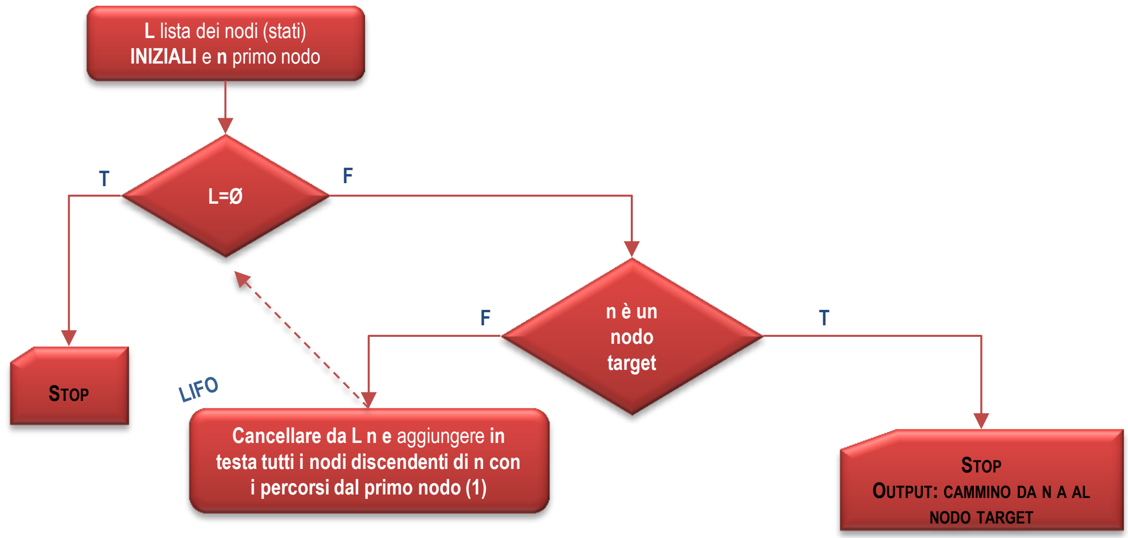
Tempo Medio di ricerca *$b^p \rightarrow O(b^p)$* $\cong$ tempo per metà dell'albero

(1) O-grande (dell'**o**rdine di), notazione matematica per comportamenti all'infinito (utile nella definizione delle complessità)
Se $f(x)=x^6+5x^3+43$ e $g(x) = x^6$, allora $f(x) \in O(g(x))$ per $x \rightarrow \infty$

Algoritmi di Ricerca di Informazioni

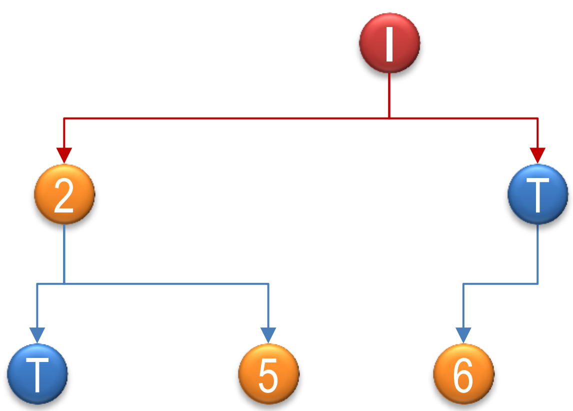
Algoritmi di visitazione degli alberi

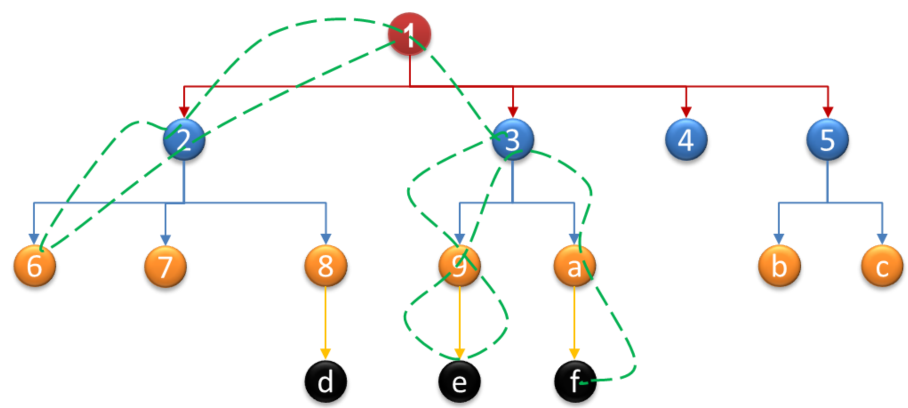
- **Ricerca in Profondità *(Depth-First)***
- *Ricerca in Larghezza (Breadth-First)*
- *Iterative Deepening Algorithm (IDA)*



Ottimalità

L'algoritmo **NON** è ottimale, in caso di più soluzione non garantisce di trovare la migliore





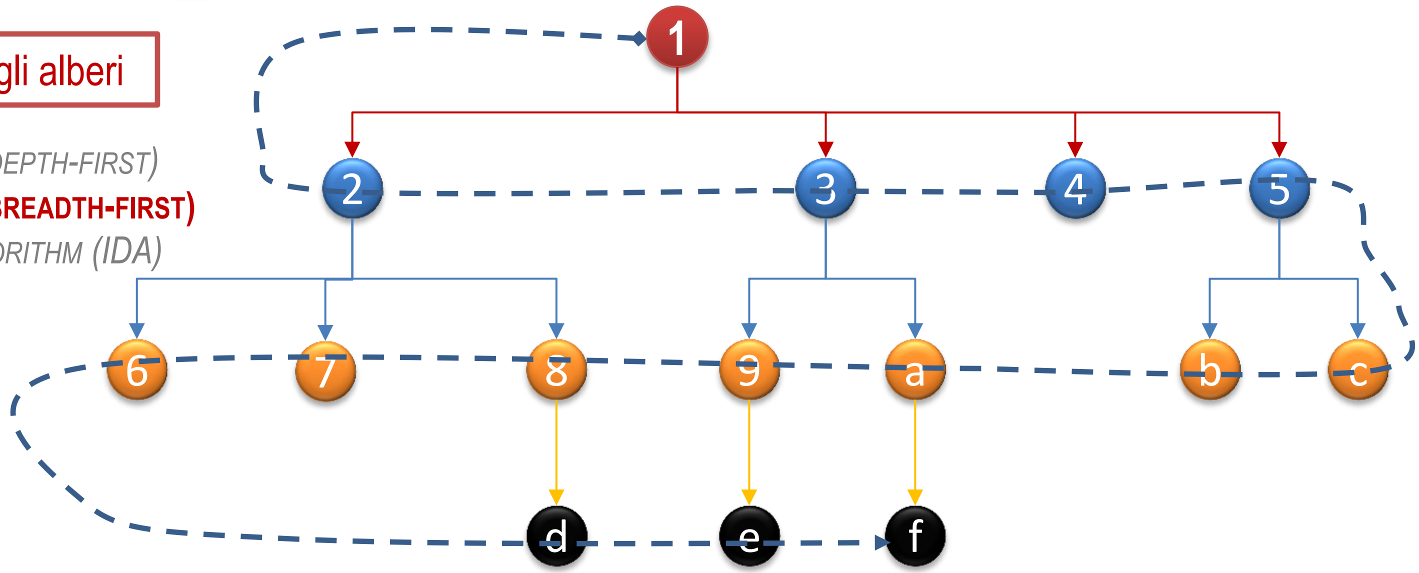
Completezza

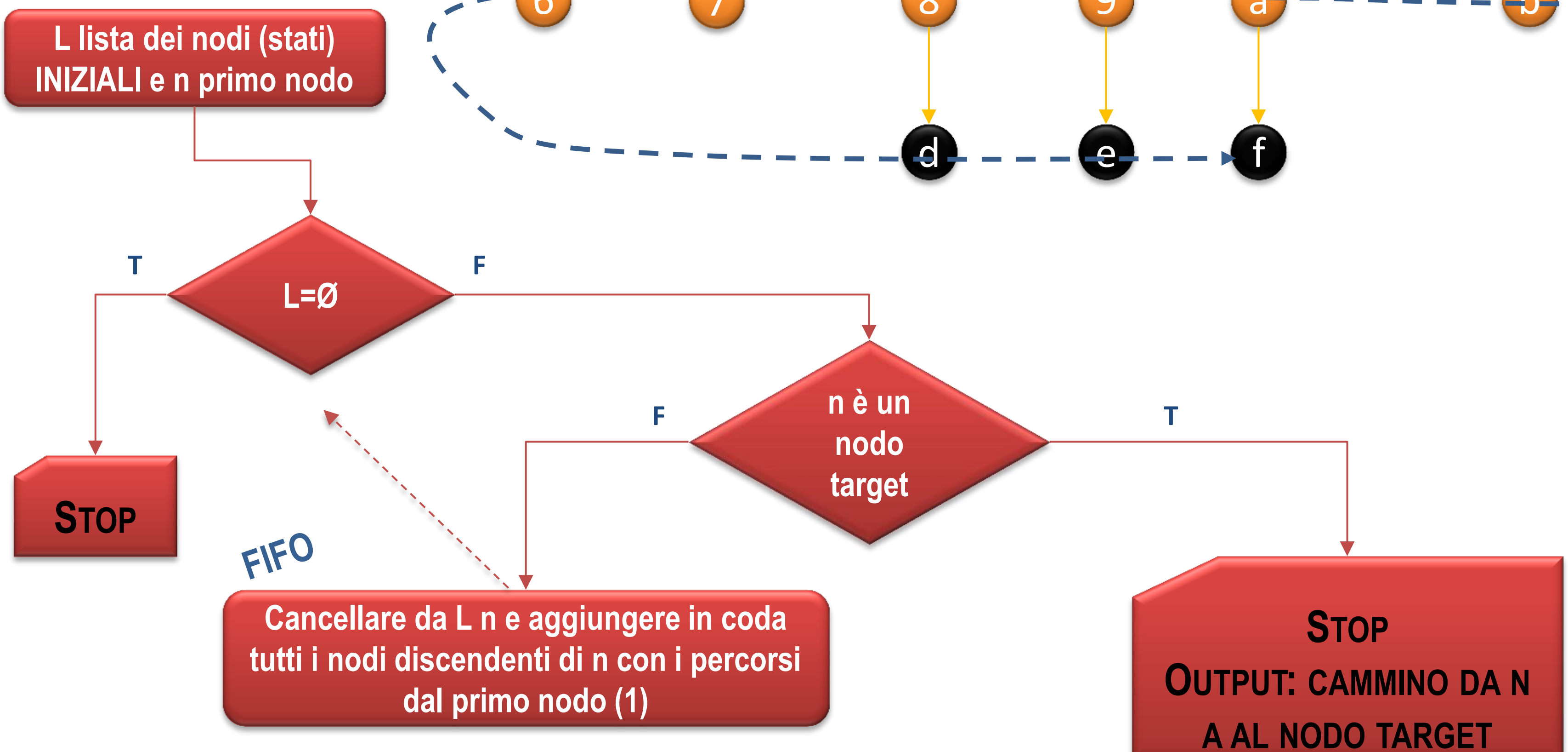
L'algoritmo **NON** è completo, in caso di alberi infiniti non garantisce il reperimento di una soluzione, se esiste.

Algoritmi di Ricerca di Informazioni

Algoritmi di visitazione degli alberi

- *Ricerca in Profondità (depth-first)*
- **Ricerca in Larghezza (breadth-first)**
- *Iterative deepening Algorithm (IDA)*

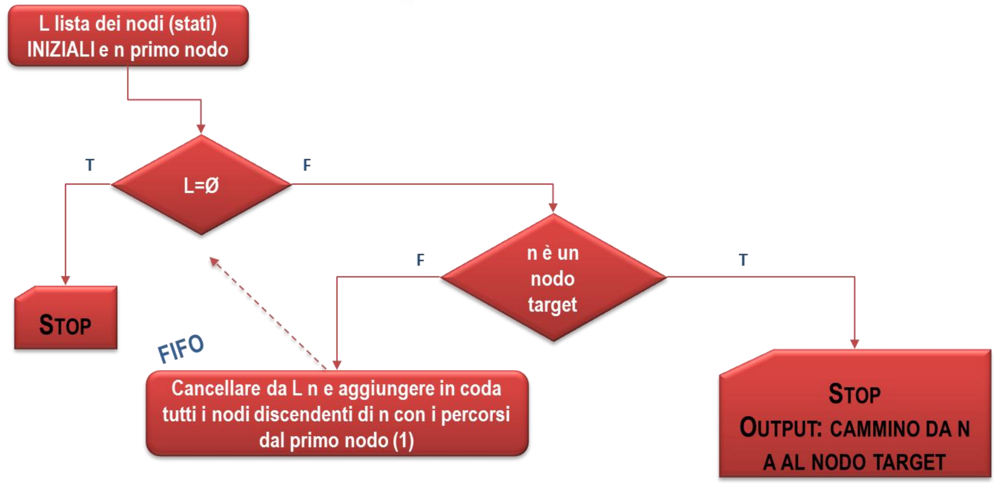




ALGORITMI DI RICERCA DI INFORMAZIONI

Algoritmi di visitazione degli alberi

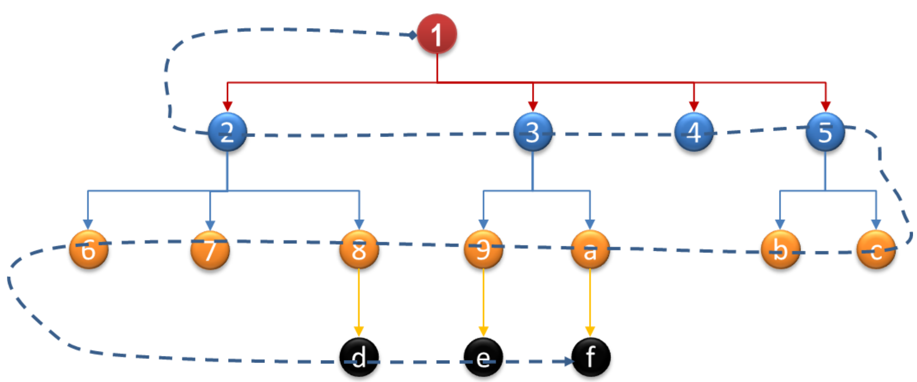
- *RICERCA IN PROFONDITÀ (DEPTH-FIRST)*
- **RICERCA IN LARGHEZZA (BREADTH-FIRST)**
- *ITERATIVE DEEPENING ALGORITHM (IDA)*



COMPLESSITÀ SPAZIALE

con **p** (*profondità*) e **b** (*bilanciamento*) ad ogni passo deve memorizzare tutti i nodi del livello **k** (quando **p=k**, si hanno $\mathbf{b^k}$ nodi)

$$b^{(p-1)} \rightarrow O(b^p)$$



COMPLESSITÀ TEMPORALE

Si deve considerare che si devono analizzare:

- Nodi intermedi (non terminali, non foglie) da analizzare prima del nodo Target a livello p è:

$$1+b+b^2+\ldots+b^{p-1}=(b^p-1)/(b-1)$$

- Numero medio di nodi di terminali (foglie) a livello p

$$(1+b^p)/2$$

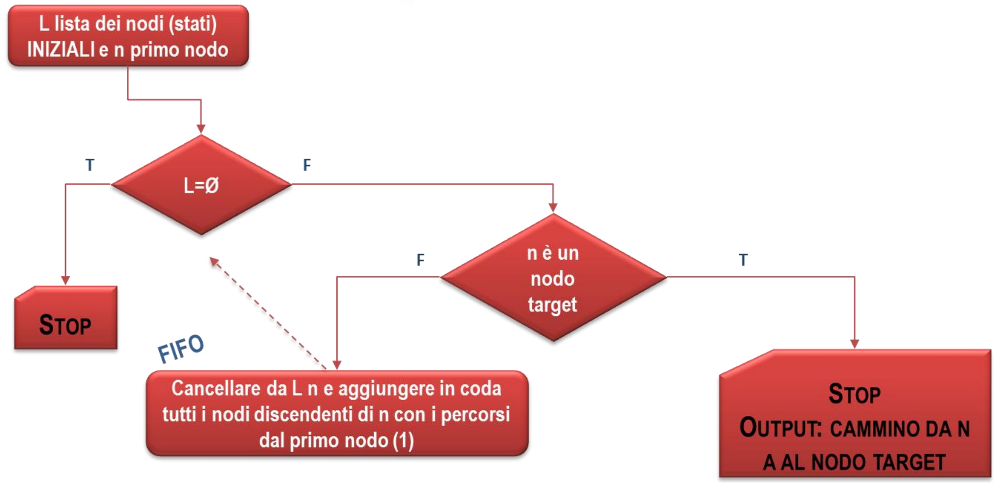
NUMERO MEDIO DI NODI (SOMMANDO)

$$b^{p+1}+b^p+b-3/2(b-1)$$

Algoritmi di Ricerca di Informazioni

Algoritmi di visitazione degli alberi

- *Ricerca in Profondità (depth-first)*
- **Ricerca in Larghezza (breadth-first)**
- *Iterative deepening Algorithm (IDA)*

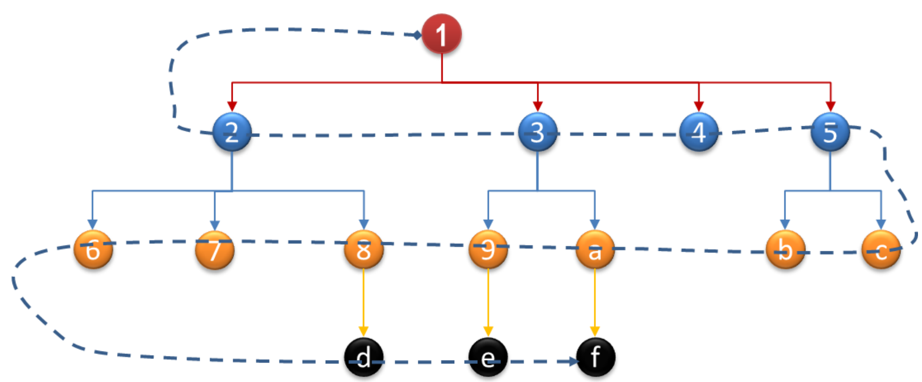


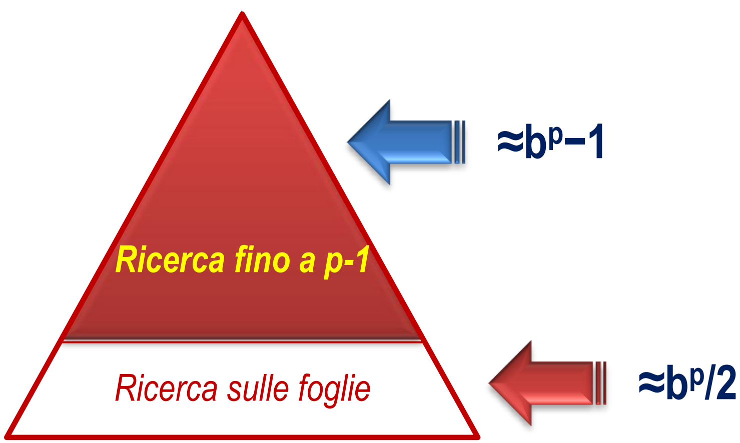
Complessità Temporale

Se **p** è molto grande (albero profondo) tempo medio di ricerca

$$b^p/2 \rightarrow O(b^p)$$

(tempo di ricerca sulle foglie)

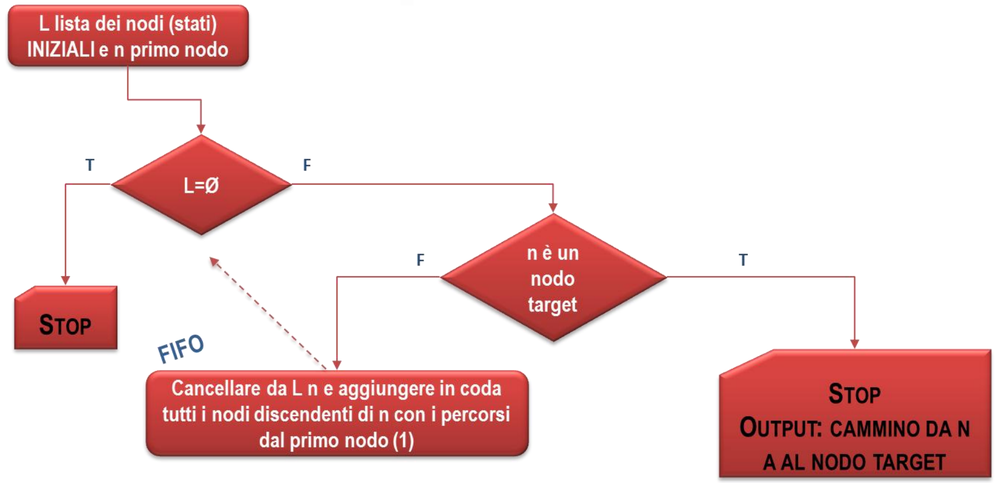




Algoritmi di Ricerca di Informazioni

Algoritmi di visitazione degli alberi

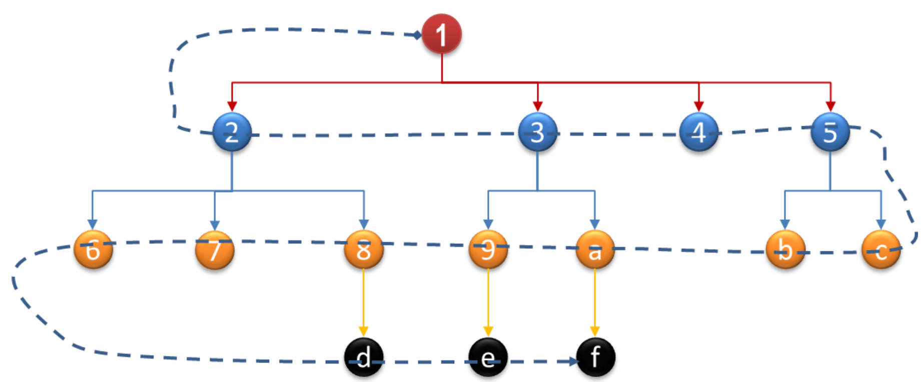
- *Ricerca in Profondità (depth-first)*
- **Ricerca in Larghezza (breadth-first)**
- *Iterative deepening Algorithm (IDA)*



La Ricerca avviene livello dopo livello.

Ottimalità

Trova sempre la soluzione migliore: quella meno profonda (p minore), ***è OTTIMALE***



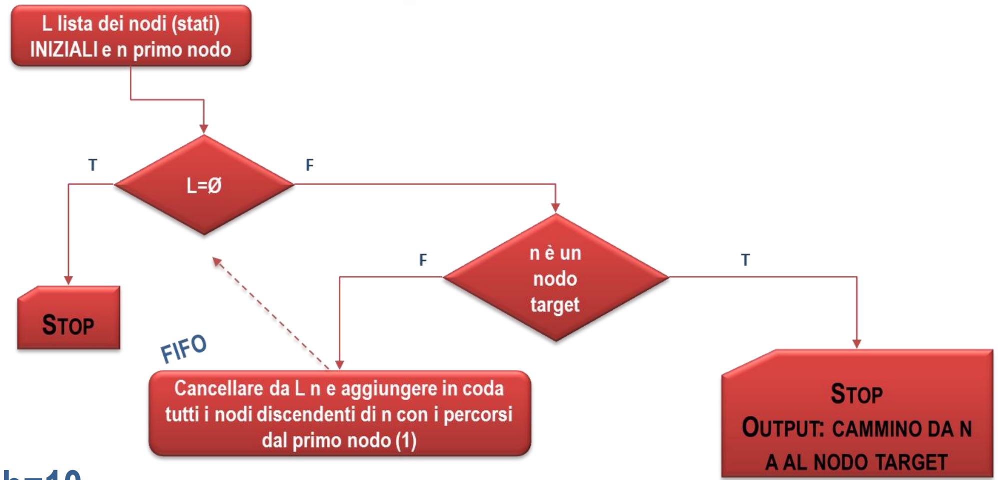
Completezza

Trova sempre la soluzione se c'è, ***è COMPLETO***

ALGORITMI DI RICERCA DI INFORMAZIONI

Algoritmi di visitazione degli alberi

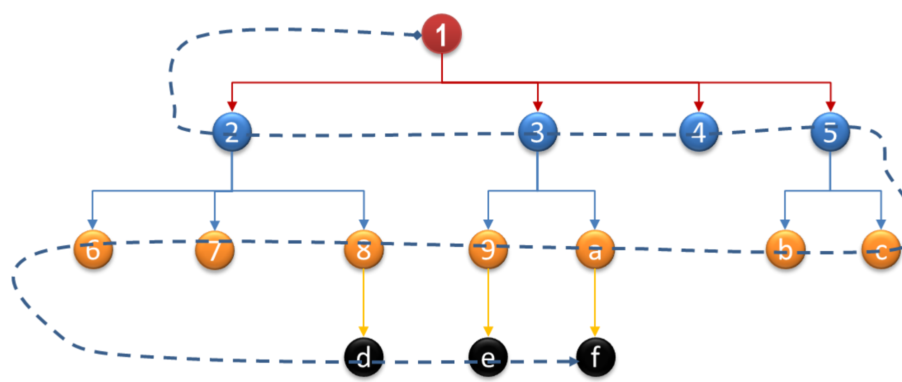
- *RICERCA IN PROFONDITÀ (DEPTH-FIRST)*
- **RICERCA IN LARGHEZZA (BREADTH-FIRST)**
- *ITERATIVE DEEPENING ALGORITHM (IDA)*



SUPPONENDO

- bilanciamento o fattore di ramificazione **b=10**
- 1.000 nodi/sec
- 100 bytes/nodo

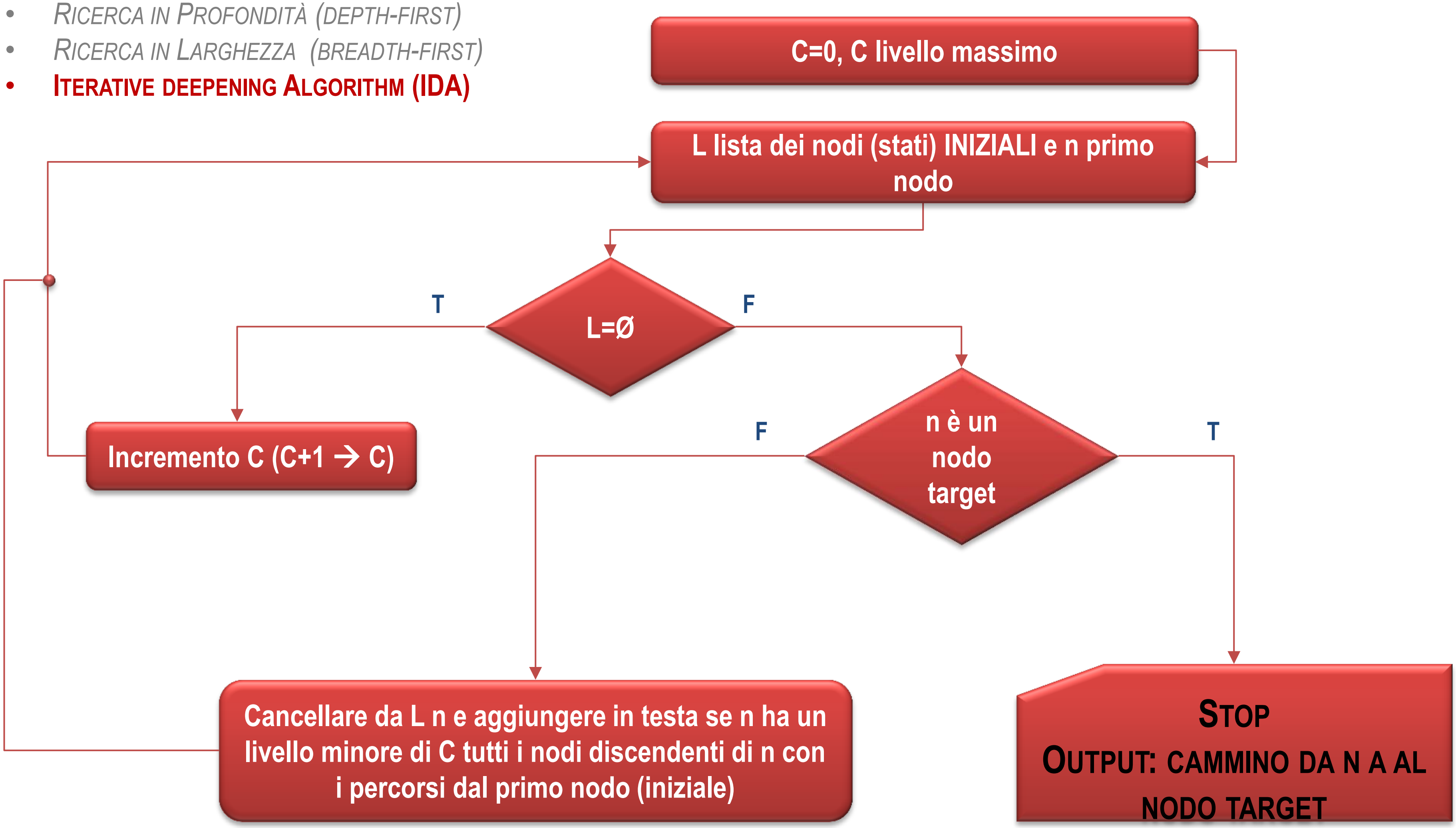
p (PROFONDITÀ)	NODI	TEMPO	MEMORIA
0	1	1^{msec}	100 b
2	1 1 1	$0{,}1^{s}$	11 Kb
4	11.111	11^{s}	1Mb
6	10^{6}	18^{m}	111 Mb
8	10^{8}	31^{h}	11 Gb
10	10^{10}	128 giorni	1 Tb
12	10^{12}	35 anni	111 Tb
14	10^{14}	3500 anni	11.111 Tb



Algoritmi di Ricerca di Informazioni

Algoritmi di visitazione degli alberi

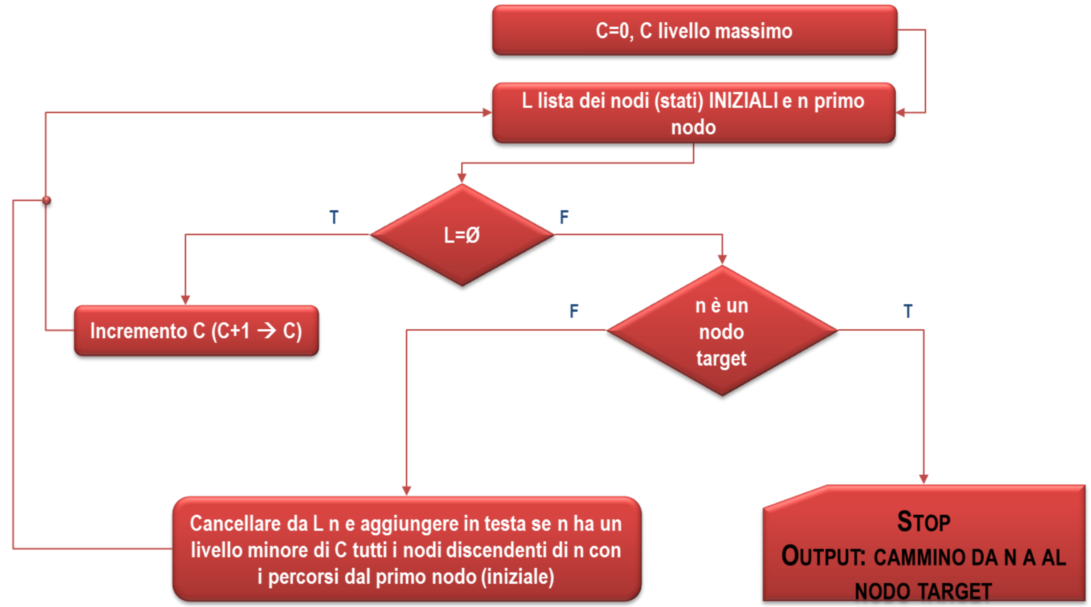
- *Ricerca in Profondità (depth-first)*
- *Ricerca in Larghezza (breadth-first)*
- **Iterative deepening Algorithm (IDA)**



Algoritmi di Ricerca di Informazioni

Algoritmi di visitazione degli alberi

- *Ricerca in Profondità (depth-first)*
- *Ricerca in Larghezza (breadth-first)*
- **Iterative deepening Algorithm (IDA)**



Complessità Spaziale

Ad ogni livello viene effettuata una ***ricerca in profondità***, serve spazio in misura di:

$p(b-1) \rightarrow O(pb)$

Complessità Temporale

Si deve considerare che:

- Nodi medio di nodi da analizzare nell'iterazione finale (livello p) è: $(b^{p-1}+bp+p-d-2)/2(b-1)$
- Numero di nodi analizzati nei livelli precedenti al livello p : $(b^{p+1}-bp-p+d)/(b-1)^2$

Sommando i Valori

$(b^{p+2}+b^{p+1}+b^2p+b^2-4bp-5b+3d+2)/2(b-1)^2$

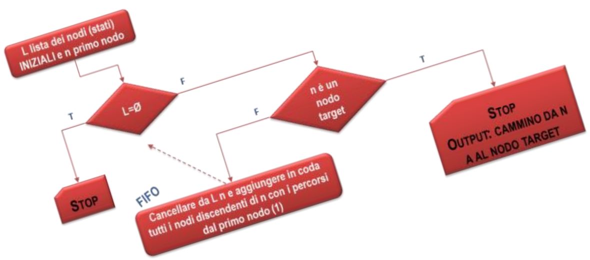
per p molto grande

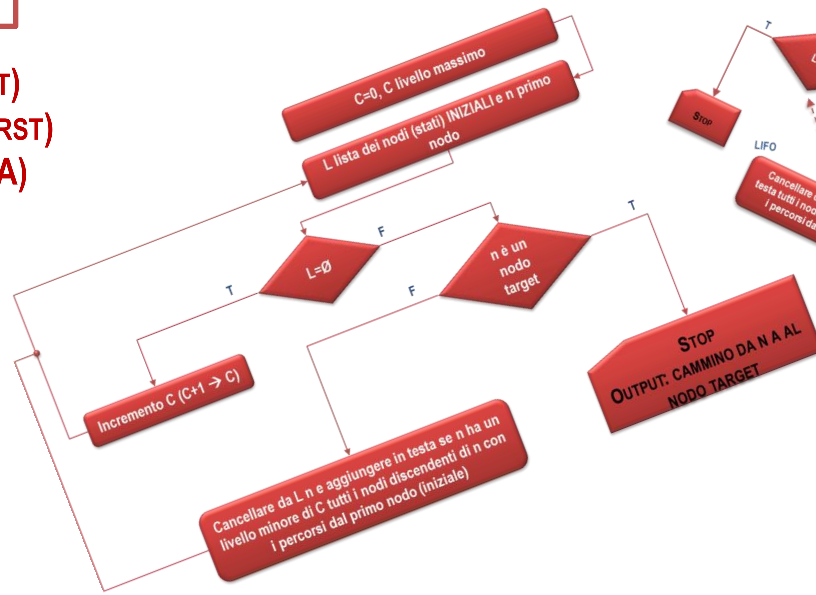
valore dominato da $(b+1)+b^{p+1})/2(b-1)^2 \rightarrow O(b^p)$

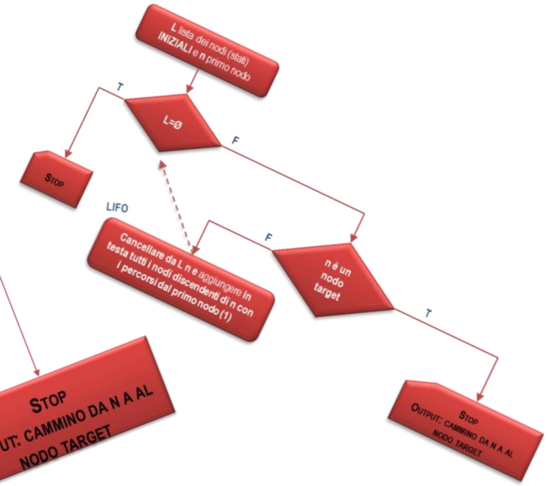
Algoritmi di Ricerca di Informazioni

Algoritmi di visitazione degli alberi

- Ricerca in Profondità (depth-first)
- Ricerca in Larghezza (breadth-first)
- Iterative deepening Algorithm (IDA)







	depth-first	breadth-first	Iterative deepening Algorithm (IDA
Complessità Spaziale	bp	b^p	bp
Complessità Temporale	b^p	b^p	b^p
Ottimalità	NO	SÌ	SÌ
Completezza	NO	SÌ	SÌ