

Πληροφορημένη αναζήτηση: A*

Μαραβίτσας Νίκος - nmaravitsas@di.uoa.gr

13 Ιανουαρίου 2014

The Tree Search Algorithm

```
function TREESearch(problem, QUEUINGFN)
returns a solution, or failure
  fringe ← MAKEQUEUE(MAKENODE(INITIALSTATE[problem]))
  loop do
    if fringe is empty then return failure
    node ← REMOVEFRONT(fringe)
    if GOALTEST[problem] applied to STATE[node] succeeds then
      return node
    fringe ← QUEUINGFN(EXPAND(node, problem), fringe)
  end
```

The Graph Search Algorithm

```
function GRAPHSEARCH(problem, QUEUINGFN)
returns a solution, or failure
  closed ← an empty set
  fringe ← MAKEQUEUE(MAKENODE(INITIALSTATE[problem]))
  loop do
    if fringe is empty then return failure
    node ← REMOVEFRONT(fringe)
    if GOALTEST[problem] applied to STATE[node] succeeds then
      return node
    if STATE[node] is not in closed then
      add STATE[node] to closed
      fringe ← QUEUINGFN(EXPAND(node, problem), fringe)
  end
```

The A* Algorithm

function BESTFIRSTSEARCH(*problem*, EVALFN)
returns a solution sequence

QueuingFn ← a function that orders nodes in
ascending order of EVALFN

return TREESearch(*problem*, *QueuingFn*)

function A*SEARCH(*problem*) **returns** a solution or failure

return BESTFIRSTSEARCH(*problem*, $g + h$)

Παράδειγμα: 8-Puzzle (με A^*)

Ευρετική: h = Αριθμός πλακιδίων σε λάθος θέση

2	8	3
1	6	4
7		5

Αρχική Κατάσταση

1	2	3
8		4
7	6	5

Κατάσταση Στόχος

Συμβολισμός: Closed, **Fringe**

Παράδειγμα: 8-Puzzle (με A*)

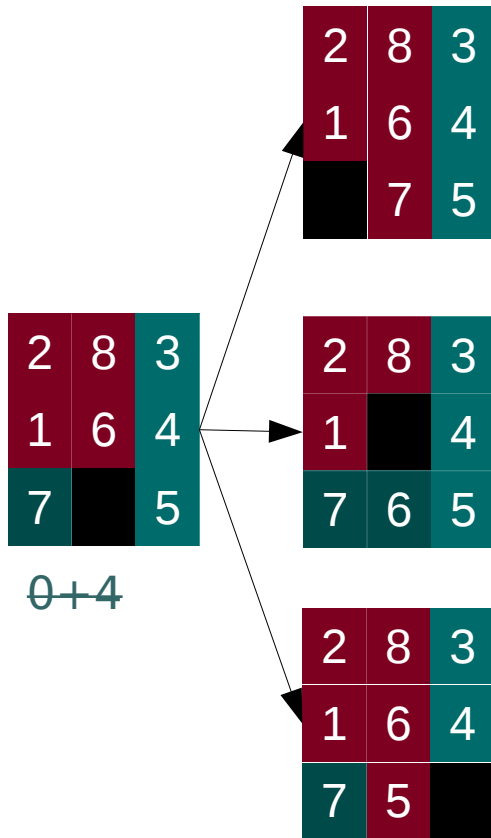
2	8	3
1	6	4
7		5

Παράδειγμα: 8-Puzzle (με A*)

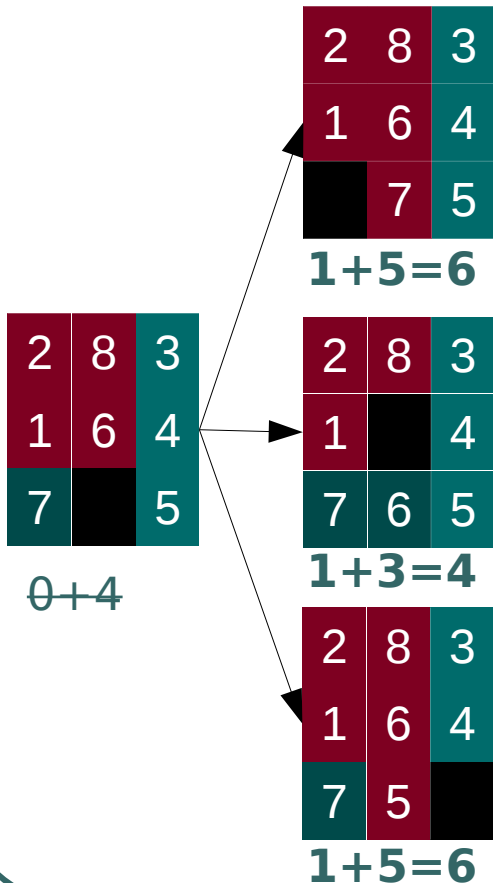
2	8	3
1	6	4
7		5

0+4

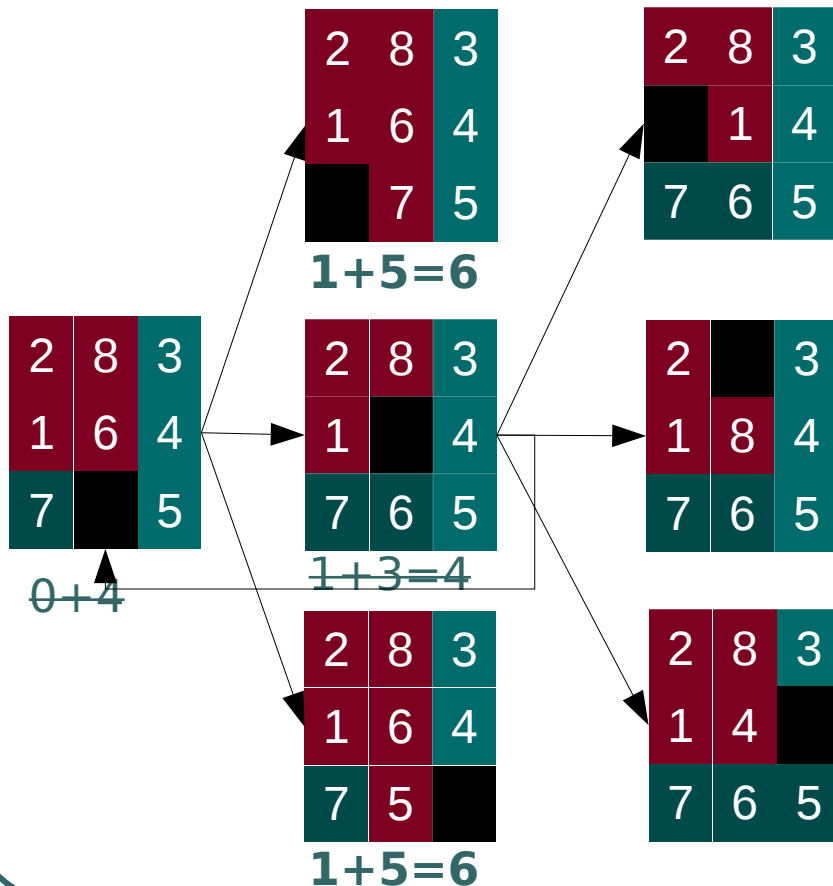
Παράδειγμα: 8-Puzzle (με A*)



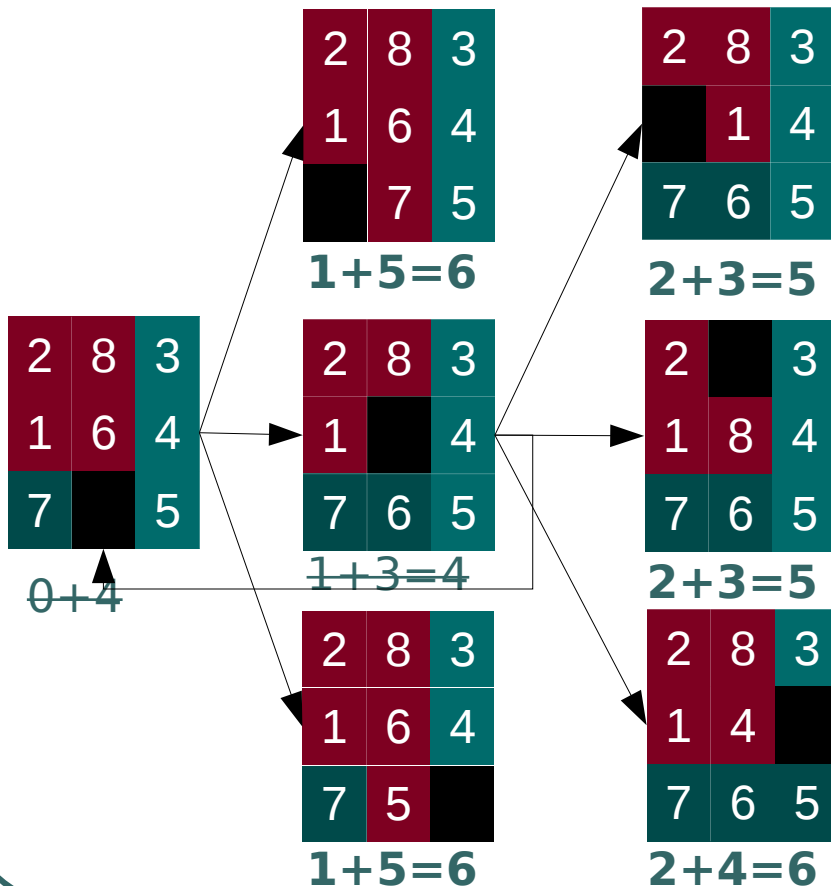
Παράδειγμα: 8-Puzzle (με A^*)



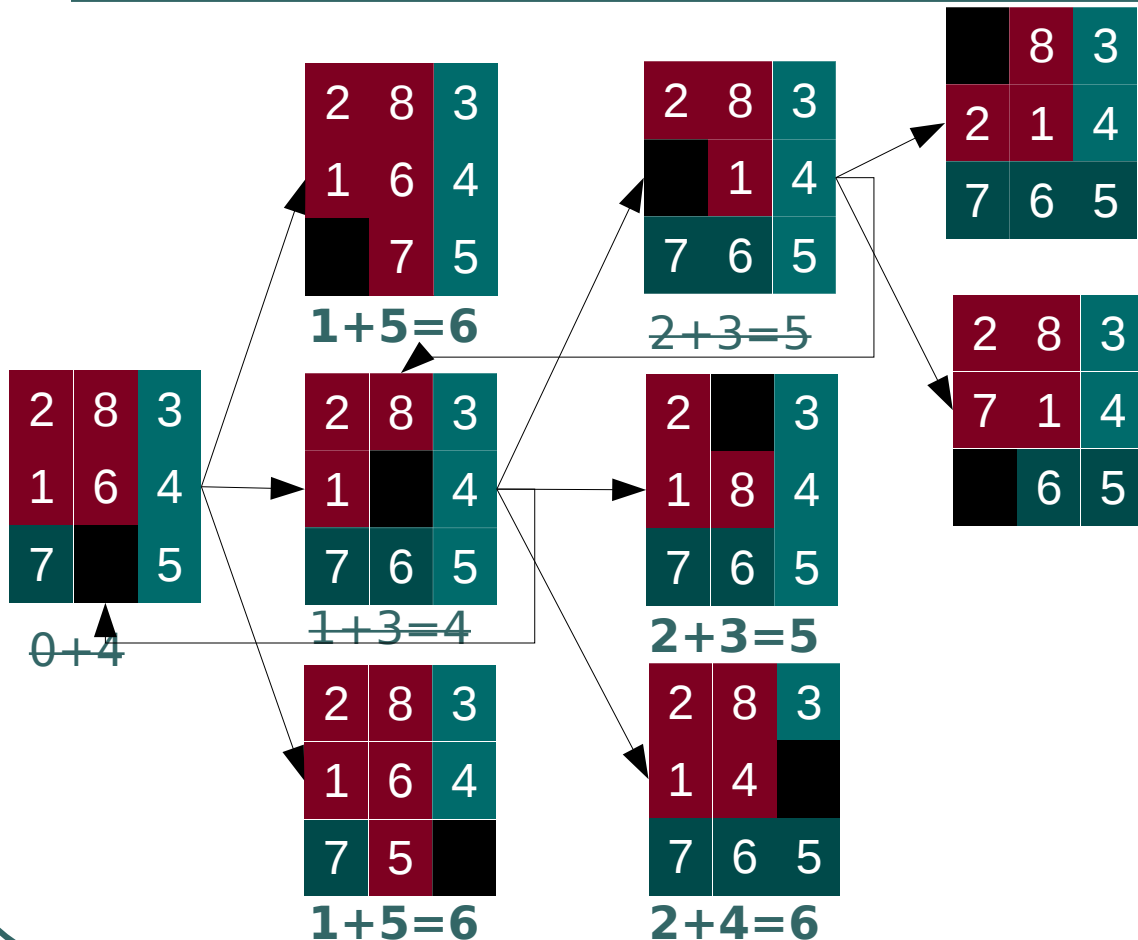
Παράδειγμα: 8-Puzzle (με A^*)



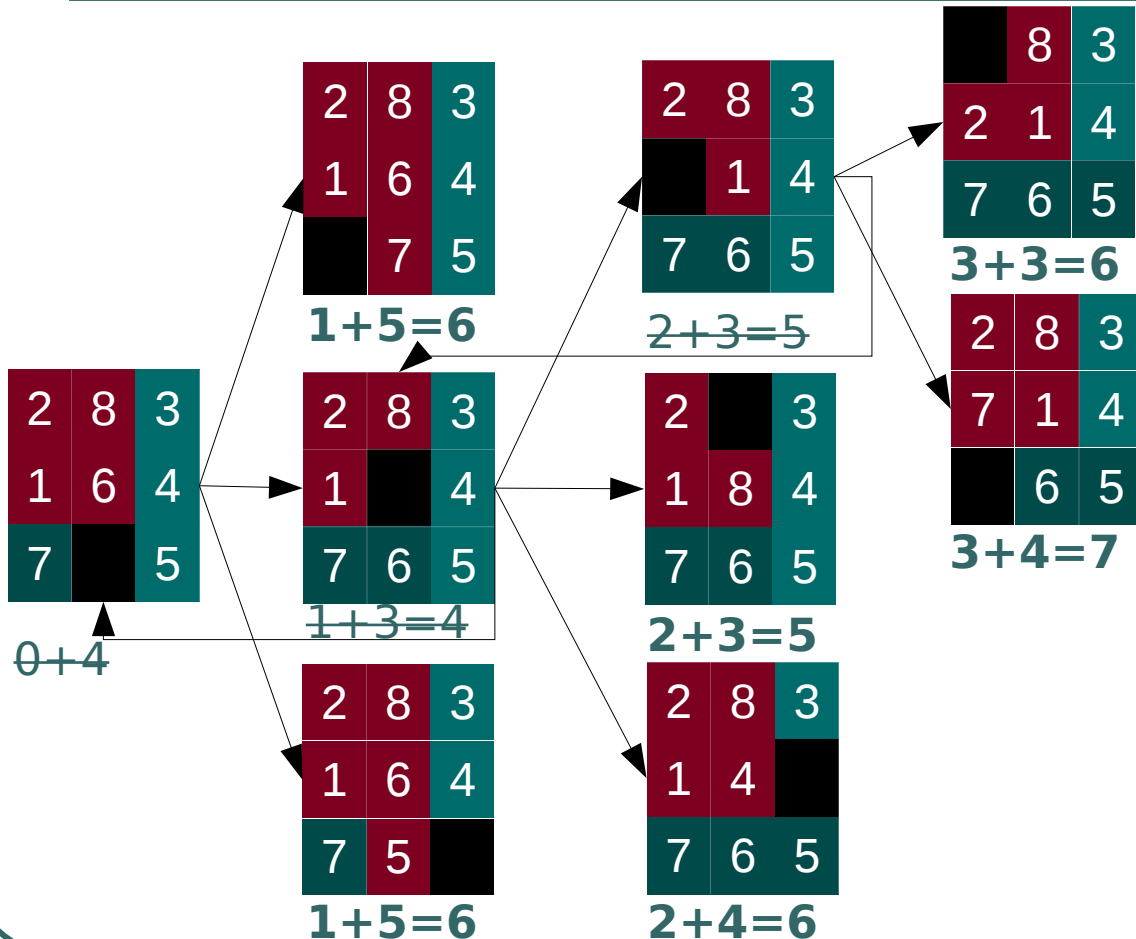
Παράδειγμα: 8-Puzzle (με A*)



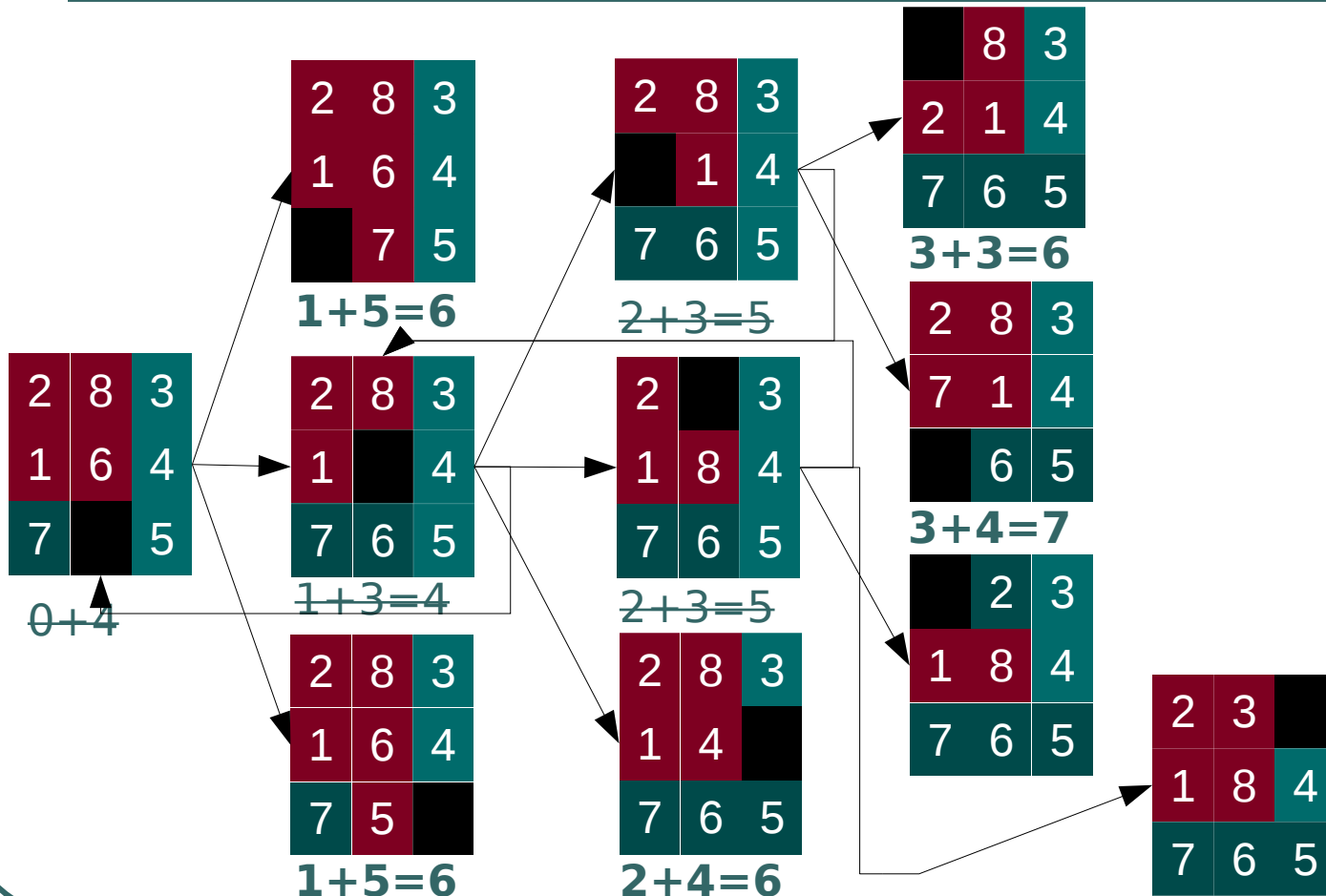
Παράδειγμα: 8-Puzzle (με A*)



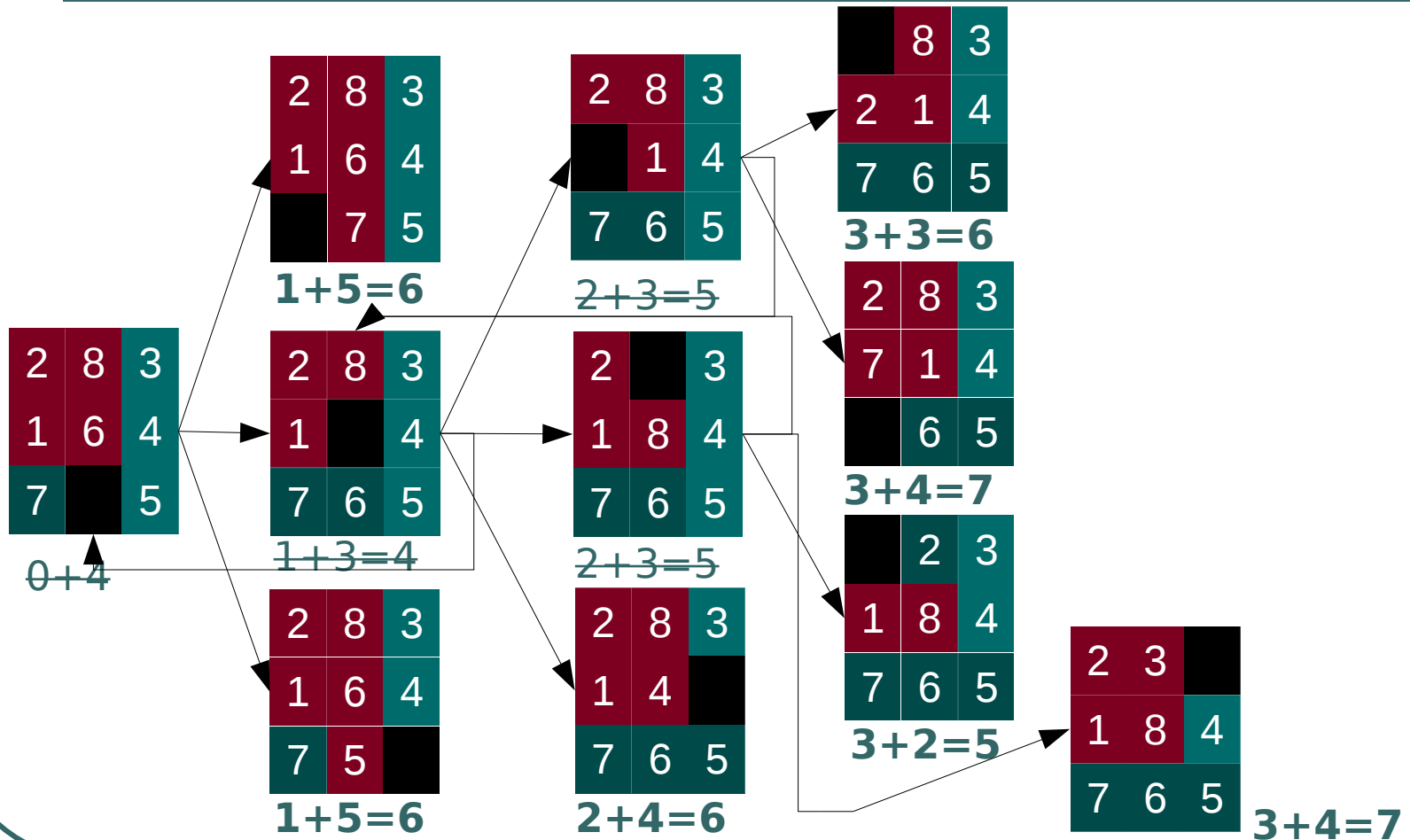
Παράδειγμα: 8-Puzzle (με A*)



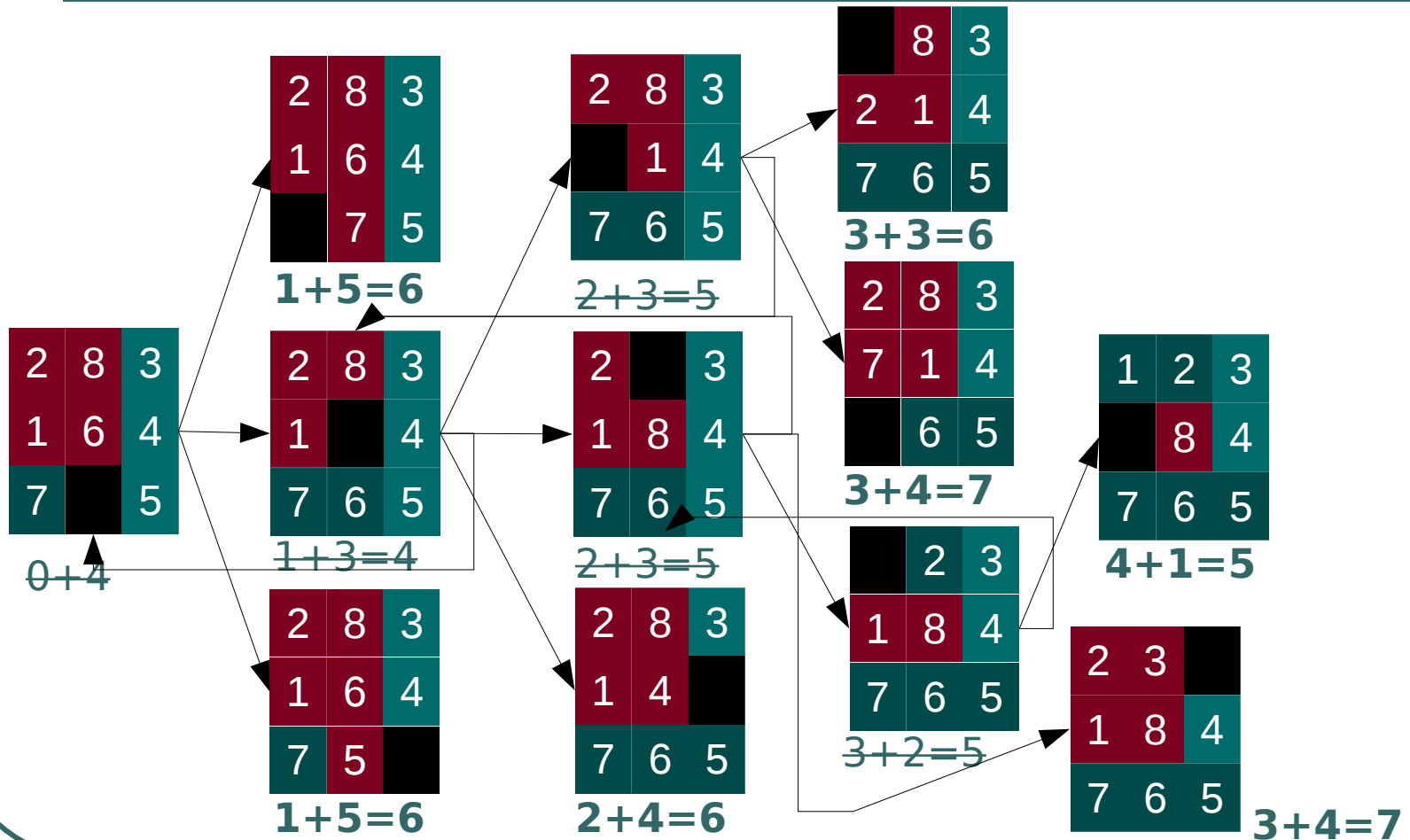
Παράδειγμα: 8-Puzzle (με A*)



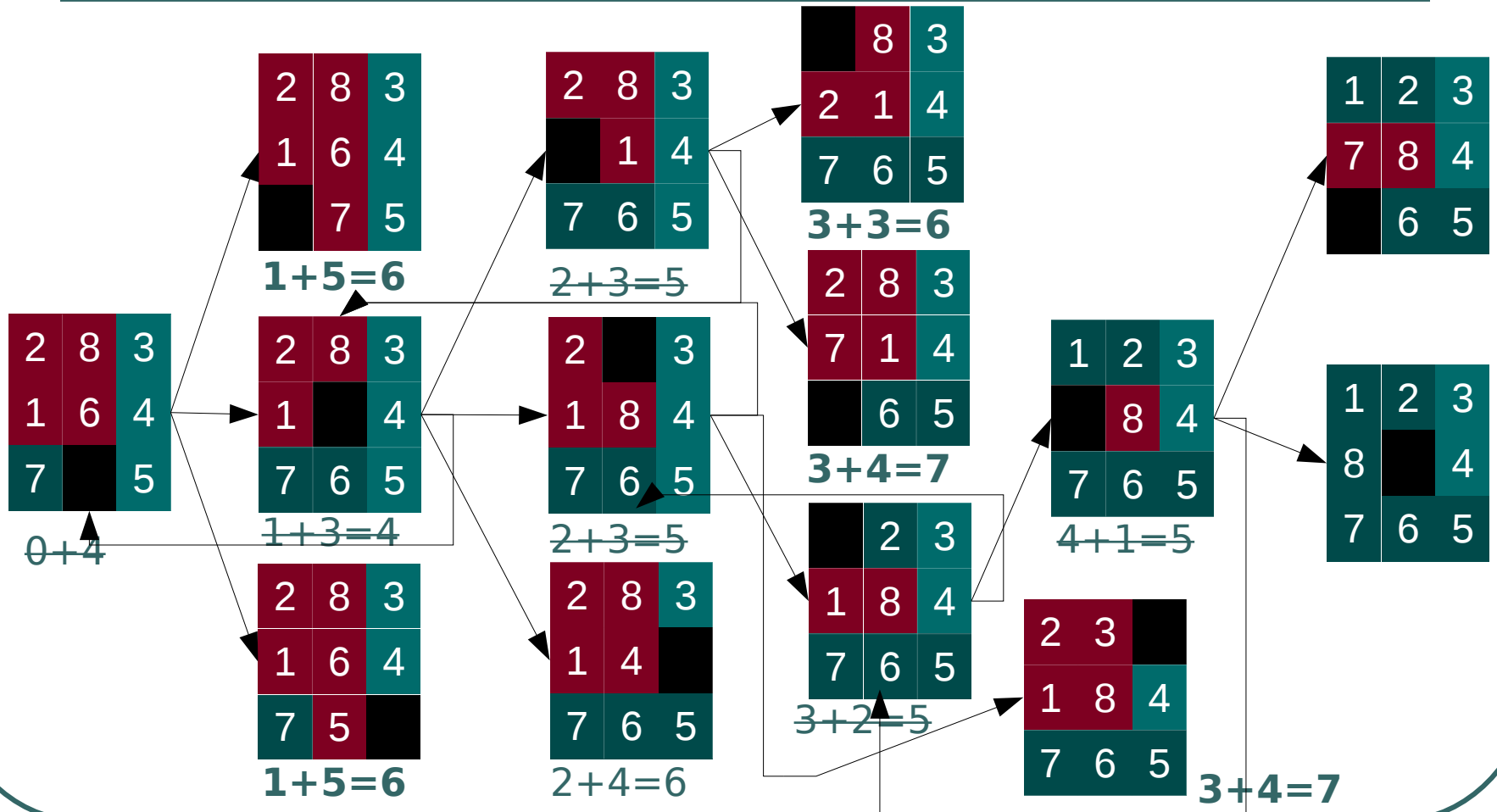
Παράδειγμα: 8-Puzzle (με A*)



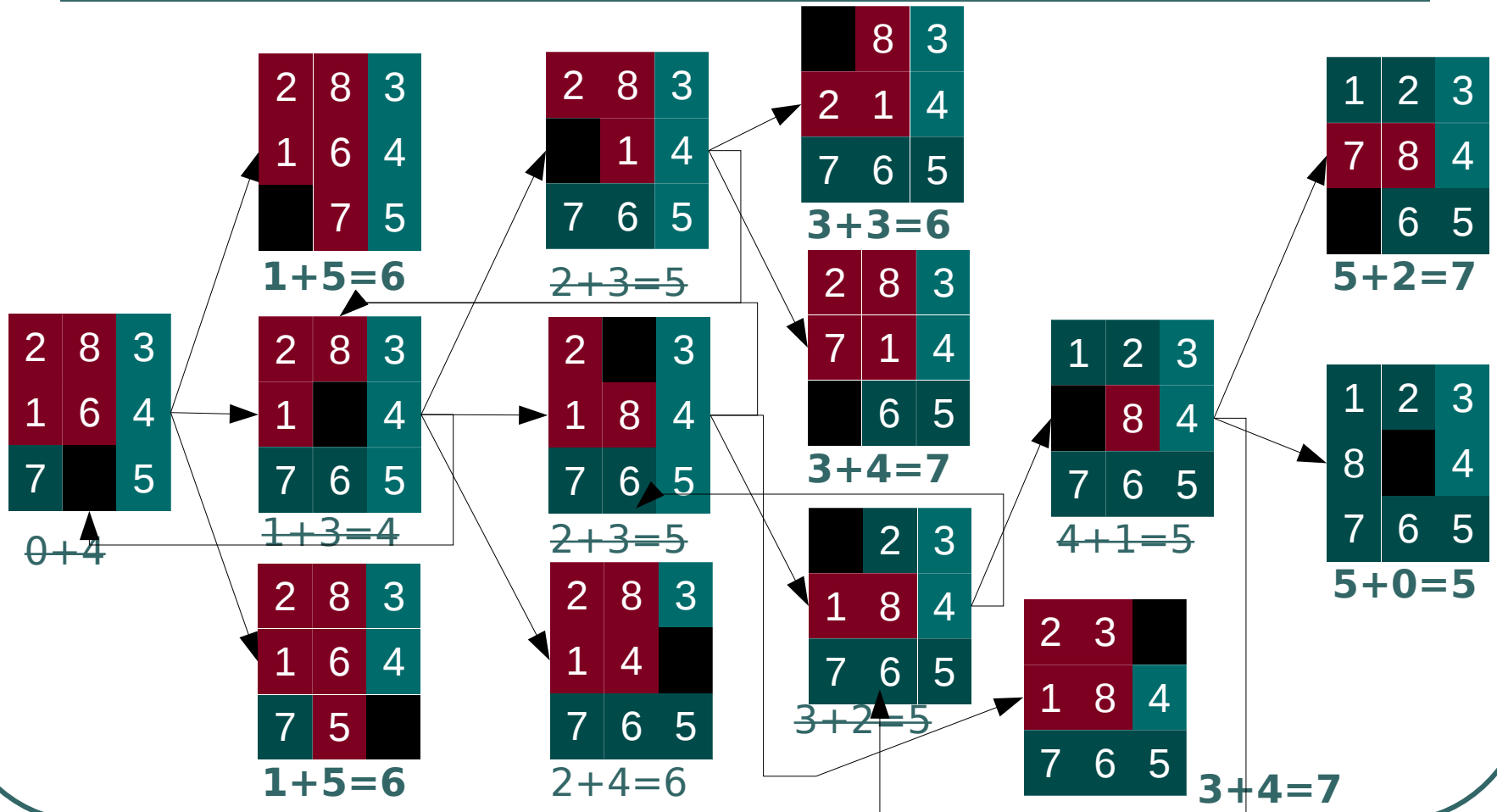
Παράδειγμα: 8-Puzzle (με A^*)



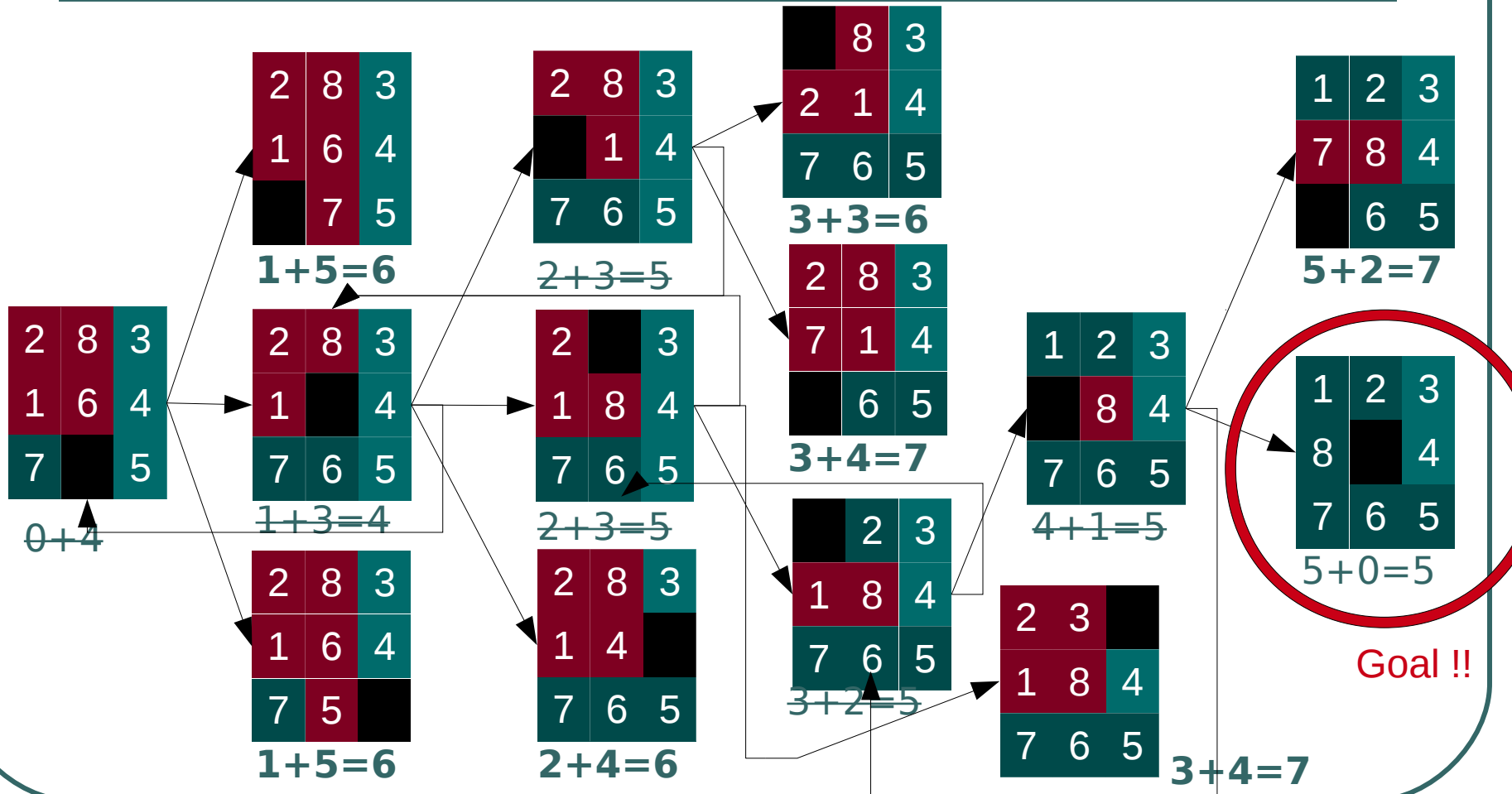
Παράδειγμα: 8-Puzzle (με A*)



Παράδειγμα: 8-Puzzle (με A*)

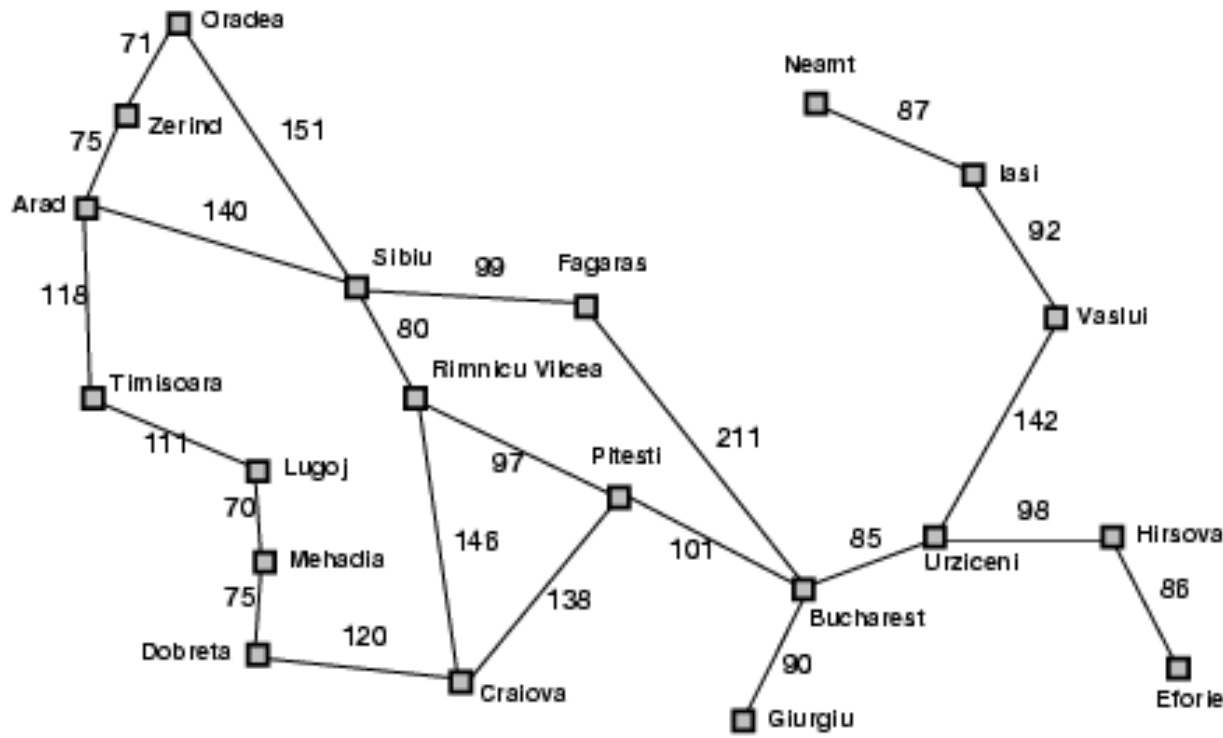


Παράδειγμα: 8-Puzzle (με A*)

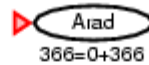


Παράδειγμα: Arad - Bucharest (με A*)

- Ευρετική: h_{SLD} = Straight Line Distance



Παράδειγμα: Arad - Bucharest



Fringe: (A, 0, None)

Closed:

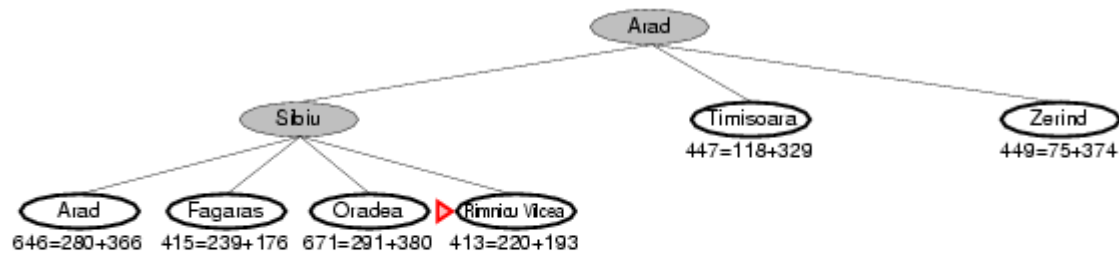
Παράδειγμα: Arad - Bucharest



Fringe: (S,393,A), (T,447,A), (Z,449,A)

Closed: A

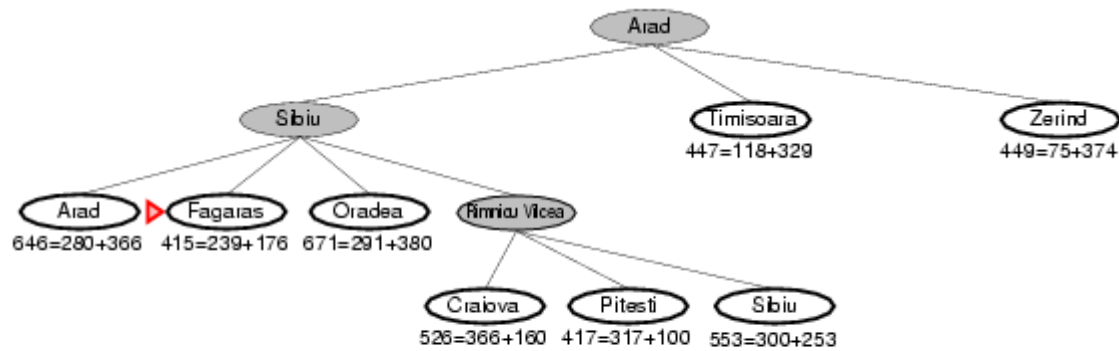
Παράδειγμα: Arad - Bucharest



Fringe: (R,413,S), (F,415,S), (T,447,A),
(Z,449,A), (A,646,S), (O,671,S)

Closed: A, S

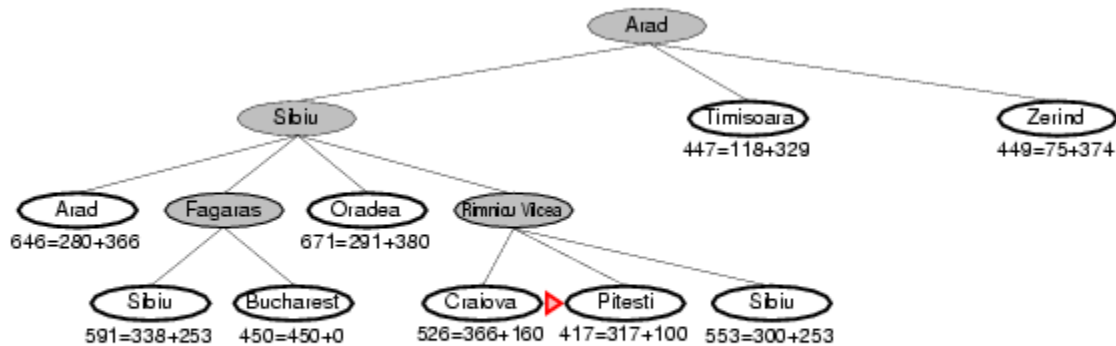
Παράδειγμα: Arad - Bucharest



Fringe: (F,415,S), (P,417,R), (T,447,A), (Z,449,A),
(C,525,R), (S,553,R), (A,646,S), (O,671,S)

Closed: A, S, R

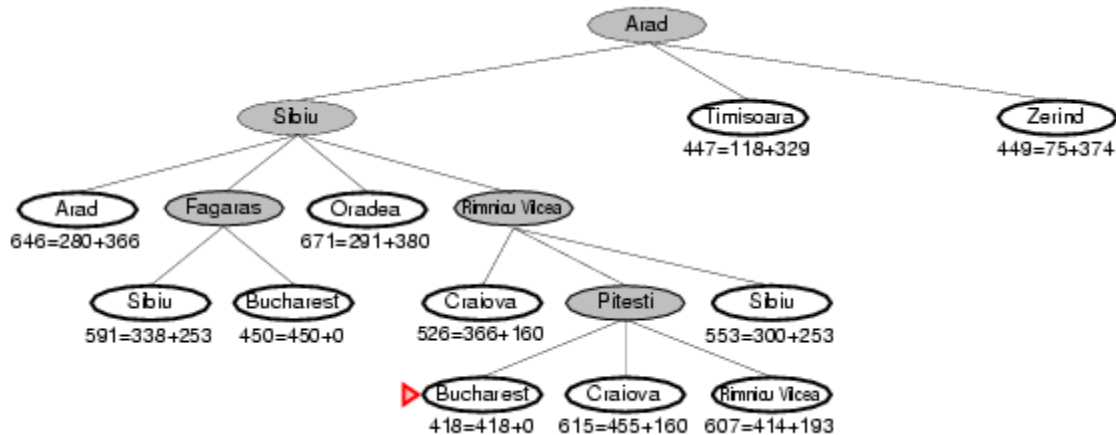
Παράδειγμα: Arad - Bucharest



Fringe: (P,417,R), (T,447,A), (Z,449,A), (B,450,F), (C,525,R), (S,553,R), (S,591,F), (A,646,S), (O,671,S)

Closed: A, S, R F

Παράδειγμα: Arad - Bucharest



Fringe: (B,418,P), (T,447,A), (Z,449,A),
 (B,450,F), (C,525,R), (S,553,R), (S,591,F),
 (R,607,P), (C,615,P), (A,646,S), (O,671,S)

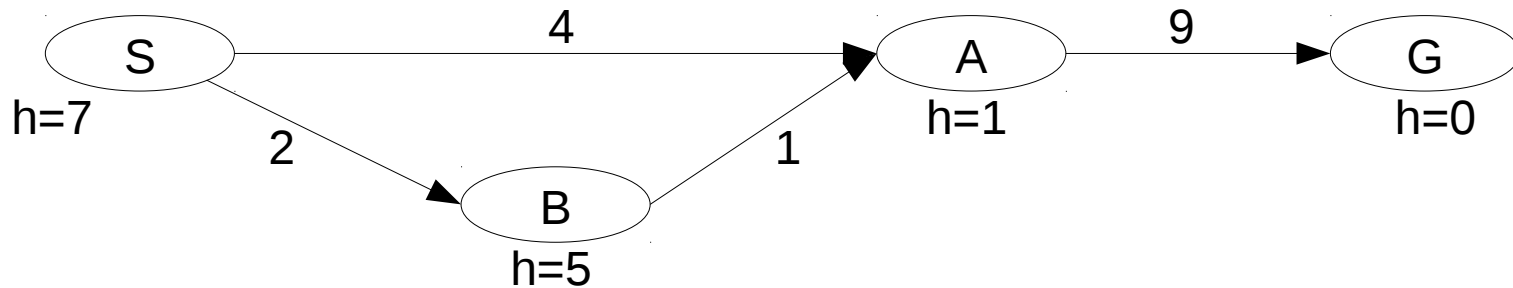
Closed: A, S, R F, P

Παράδειγμα: Arad - Bucharest

- Χρησιμοποιήσαμε καλή ευρετική;
- Παραδεκτή: η μικρότερη απόσταση είναι η ευθεία
- Συνεπής: $h(n) \leq c(n, a, n') + h(n')$

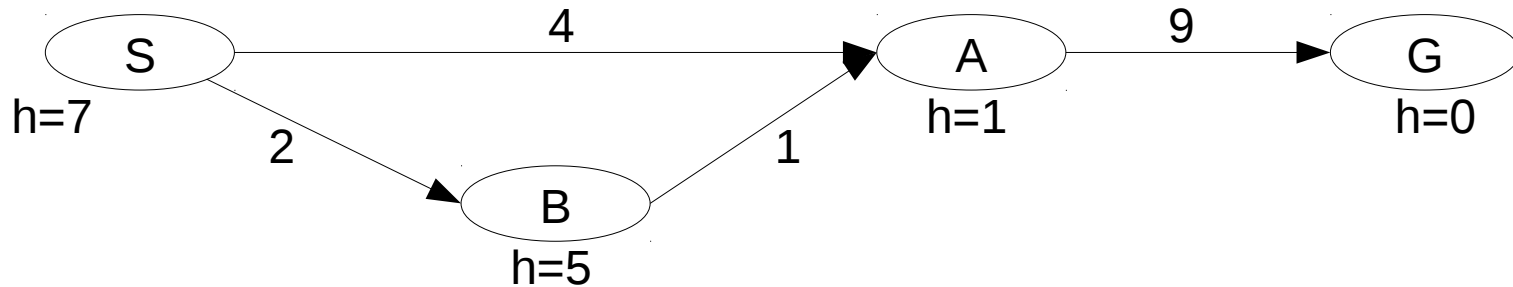
Τριγωνική ανισότητα

Παράδειγμα ασυνεπούς ευριστικής



$h(B) \leq c(B,A) + h(A)$ -- ΔΕΝ ΙΣΧΥΕΙ !

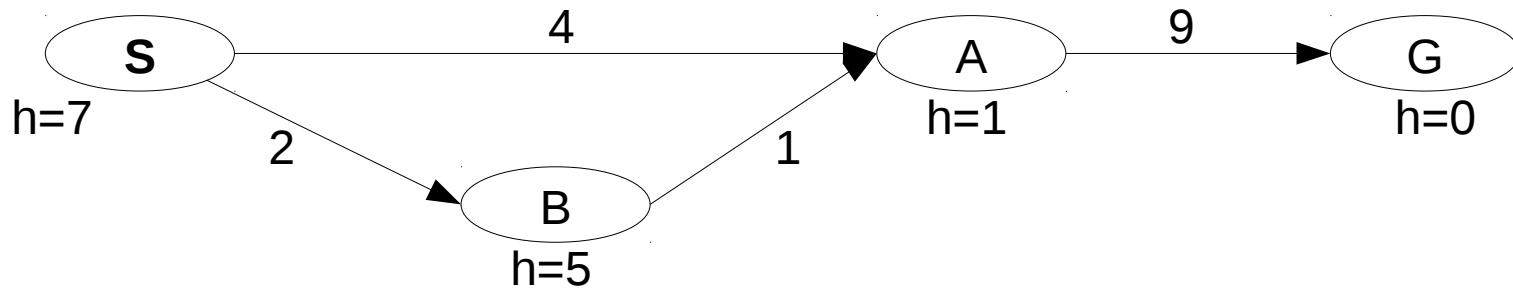
Παράδειγμα ασυνεπούς ευριστικής



Fringe: (S,0,None)

Close:

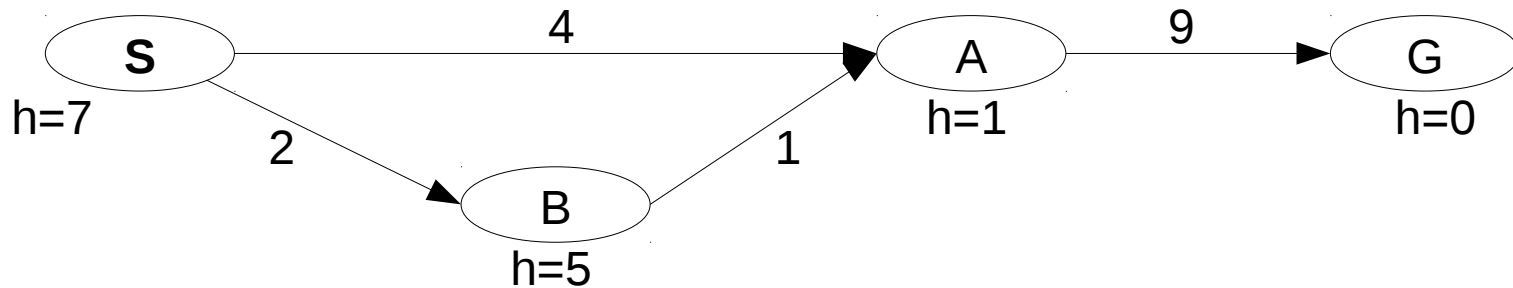
Παράδειγμα ασυνεπούς ευριστικής



Fringe: (A,5,S), (B,7,S)

Close: S

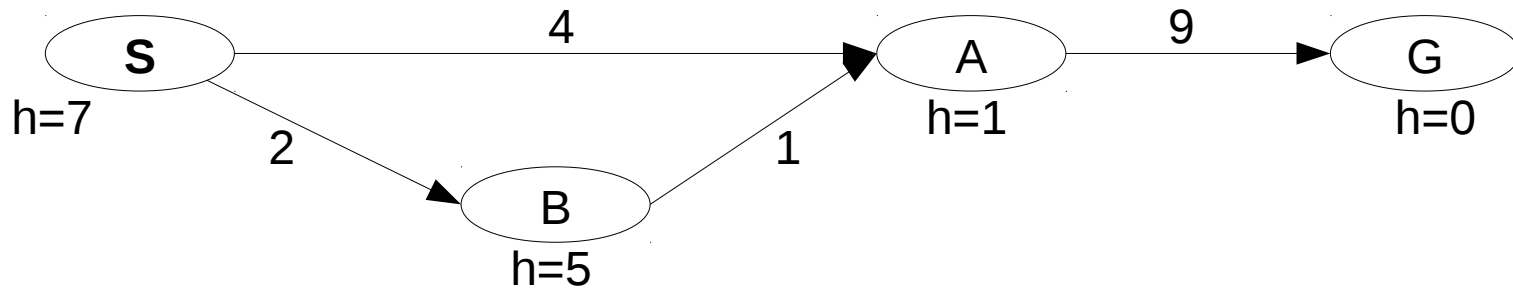
Παράδειγμα ασυνεπούς ευριστικής



Fringe: (B, 7, S), (G, 13, A)

Close: S, A

Παράδειγμα ασυνέπους ευριστικής

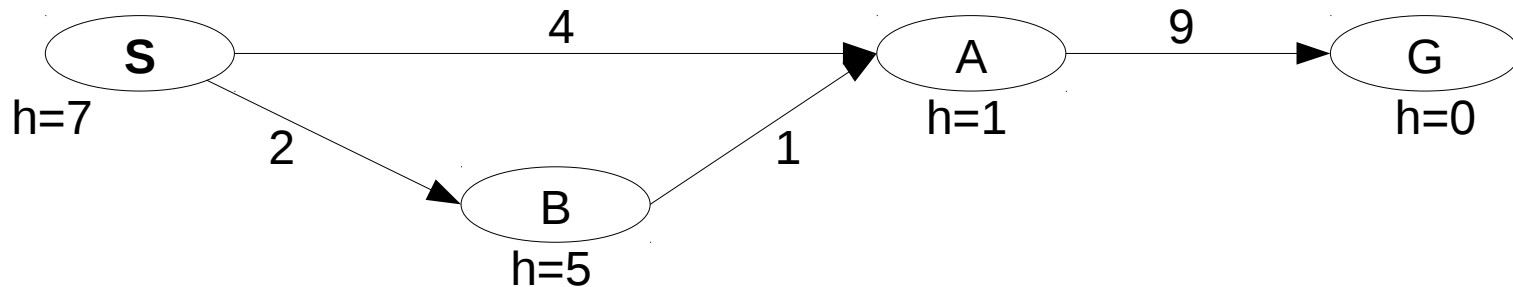


Fringe: (A, 4, B)*, (G, 13, A)

Close: S, A, B

* Δεν επεκτείνεται, γιατί το A είναι στο σύνολο *Close*

Παράδειγμα ασυνέπους ευριστικής



Fringe: (G, 13, A)

Close: S, A, B

Οπότε σαν λύση βγαίνει το μονοπάτι $S \rightarrow A \rightarrow G$,
με κόστος 13 αντί του μονοπατιού $S \rightarrow B \rightarrow A \rightarrow G$
με κόστος 12.

Ευριστικές για 8-puzzle

Ένα πλακίδιο μπορεί να μετακινηθεί από ένα τετράγωνο A στο τετράγωνο B αν

1. το A συνορεύει με το B
2. Το B είναι κενό

Αφαιρώντας μία (ή και τις δύο συνθήκες) προκύπτουν **παραδεκτές** και **συνεπής** ευριστικές:

- $-(1)$: απόσταση manhattan
- $-(2)$: μηχανισμός gashing
- $-(1+2)$: πλήθος πλακιδίων που σε λάθος θέση

Μηχανισμός Gaschnig

Μέχρι να επιτευχθεί η κατάσταση στόχου:

1. Αν το κενό πλακίδιο είναι σε λάθος θέση μετέφερε σ' αυτή τη θέση το σωστό
2. Αλλιώς μετέφερε οποιοδήποτε πλακίδιο που είναι σε λάθος θέση στη θέση του κενού
3. Για κάθε βήμα μέτρα 1

Ποια ευρετική είναι η καλύτερη;

Παράδειγμα ($h_G > h_M$)

1	2	3
4	8	6
7	5	

Misplaced tiles: 2
Manhattan distance: 2
Gaschnig: 3

Παράδειγμα ($h_G < h_M$)

1	2	3
4	5	8
7	6	

Misplaced tiles: 2
Manhattan distance: 4
Gaschnig: 3

Λύση: $h = \max(h_G, h_M)$

Βιβλιογραφία

- Τεχνητή Νοημοσύνη: Μια σύγχρονη προσέγγιση – Stuart Russel, Peter Norving