



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

Τεχνητή νοημοσύνη και μηχανική μάθηση

Εισηγητής
Αναστάσιος Κεσίδης

Επίλυση προβλημάτων

Εισαγωγή

➤ Παράδειγμα

Μετακίνηση από το Arad προς το Βουκουρέστι



Τερματισμός

Διατύπωση

➤ Διατύπωση στόχου (goal formulation)

Στόχος

Ένα σύνολο καταστάσεων του κόσμου στις οποίες ο στόχος εκπληρώνεται



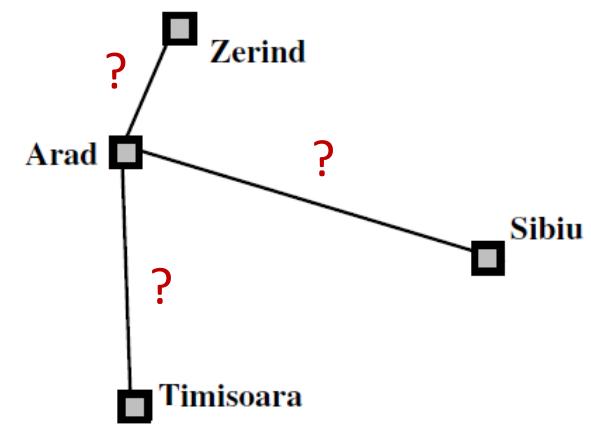
➤ Διατύπωση προβλήματος (problem formulation)

Επιλογή ενεργειών και καταστάσεων που πρέπει να εξεταστούν με δεδομένο έναν στόχο.

Έλλειψη πληροφορίας σχετικά με την κατάσταση που προκύπτει μετά από κάθε ενέργεια;



Ενέργειες στην τύχη



Συνθήκες

➤ Υποθέσεις για το περιβάλλον

- Είναι **παρατηρήσιμο**

Η τρέχουσα κατάσταση είναι γνωστή

- Είναι **διακριτό**

Υπάρχει ένα πεπερασμένο πλήθος πιθανών ενεργειών

- Είναι **γνωστό**

Είναι σαφές σε ποια κατάσταση οδηγεί η κάθε ενέργεια

- Το περιβάλλον είναι **αιτιοκρατικό** (deterministic)

Κάθε ενέργεια οδηγεί σε μια συγκεκριμένη κατάσταση

Υπό τις παραπάνω συνθήκες:

Η λύση ενός προβλήματος είναι μια **καθορισμένη ακολουθία ενεργειών**

Διατύπωση



Αναζήτηση



Εκτέλεση

Καλώς ορισμένα προβλήματα

➤ Αρχική κατάσταση

Η κατάσταση στην οποία ξεκινά η επίλυση του προβλήματος

Π.χ.

$$In(Arad)$$

➤ Ενέργειες

Δεδομένης μιας αρχικής κατάστασης s το σύνολο $ACTIONS(s)$ δίνει το τις ενέργειες που μπορούν να εκτελεστούν στην κατάσταση αυτή.

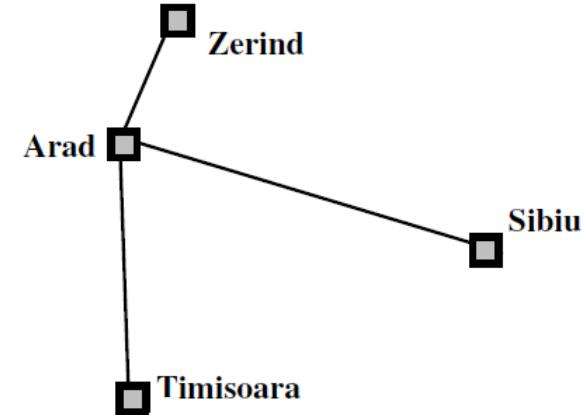
Π.χ.

για την κατάσταση s

$$In(Arad)$$

το σύνολο ενεργειών $ACTIONS(s)$ είναι

$$\begin{aligned} ACTIONS(In(Arad)) \\ = \{Go(Zerind), Go(Sibiu), Go(Timisoara)\} \end{aligned}$$



Καλώς ορισμένα προβλήματα

➤ Μοντέλο μετάβασης

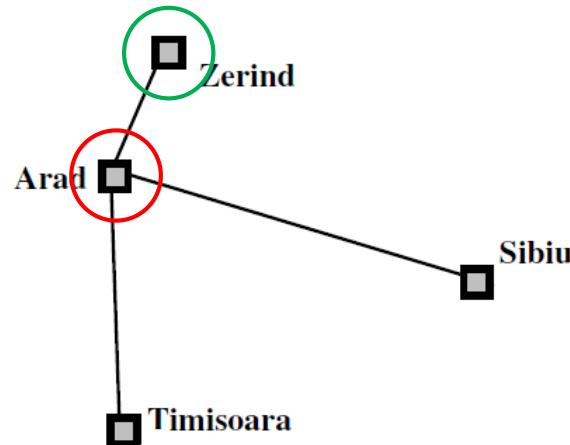
Η Συνάρτηση διαδόχων

$$Result(s, a)$$

δίνει την διάδοχη κατάσταση που προκύπτει όταν εφαρμοστεί
η ενέργεια a στην κατάσταση s

Π.χ.

$$RESULT(In(Arad), Go(Zerind)) = In(Zerind)$$



Καλώς ορισμένα προβλήματα

➤ Χώρος καταστάσεων

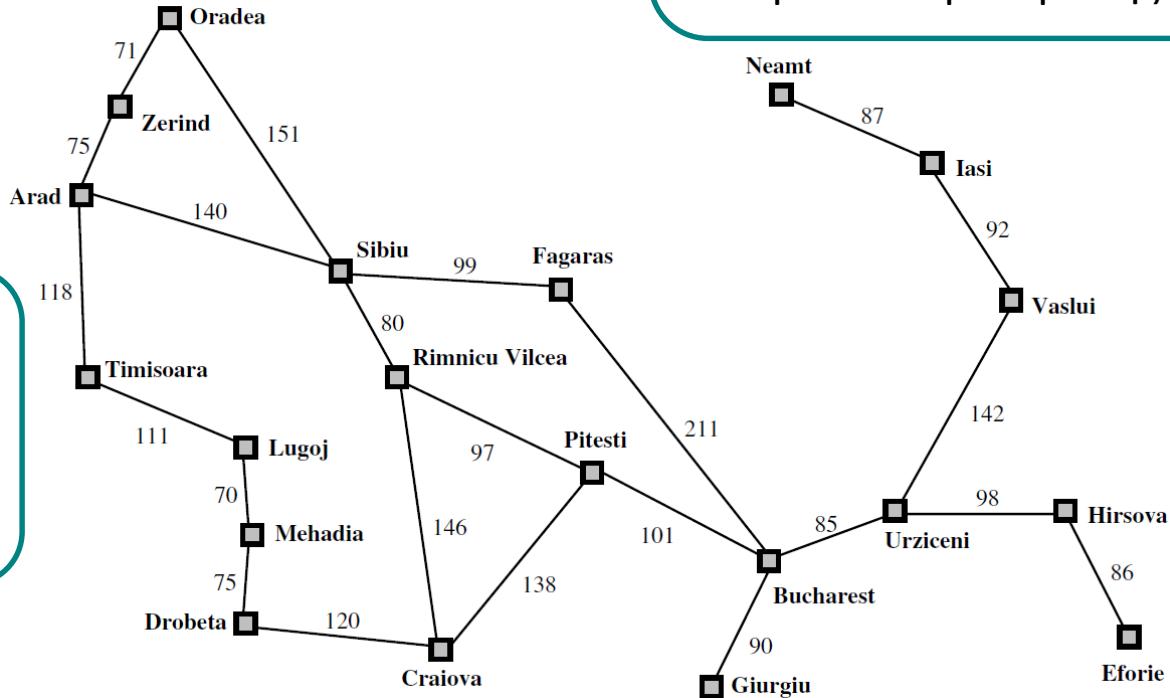
Το σύνολο όλων των καταστάσεων προσπελάσιμες από την αρχική κατάσταση

Απεικόνιση μέσω γράφου όπου

- οι κόμβοι είναι καταστάσεις
- οι ακμές είναι οι ενέργειες

Διαδρομή

Ακολουθία καταστάσεων συνδεδεμένων μέσω ακολουθίας ενεργειών



Χώρος καταστάσεων

=

Αρχική κατάσταση
+ ενέργειες
+ μοντέλο μετάβασης

Καλώς ορισμένα προβλήματα

➤ Έλεγχος στόχου

Έλεγχος εάν η τρέχουσα κατάσταση είναι η **κατάσταση-στόχος** του προβλήματος

Μπορεί να είναι

- **Επακριβώς καθορισμένο σύνολο**

Ένα υποσύνολο του χώρου καταστάσεων

Π.χ.

$$\{In(Bucharest)\}$$

- **Ιδιότητα**

Η ιδιότητα μιας κατάστασης η οποία αντιστοιχεί στην επίτευξη του στόχου

Π.χ.

Ρου-Ματ στο σκάκι (ο αντίπαλος βασιλιάς δεν μπορεί κάνει καμία κίνηση)

Καλώς ορισμένα προβλήματα

➤ Συνάρτηση κόστους διαδρομής

Μια συνάρτηση που αποδίδει **αριθμητική τιμή κόστους** σε κάθε διαδρομή

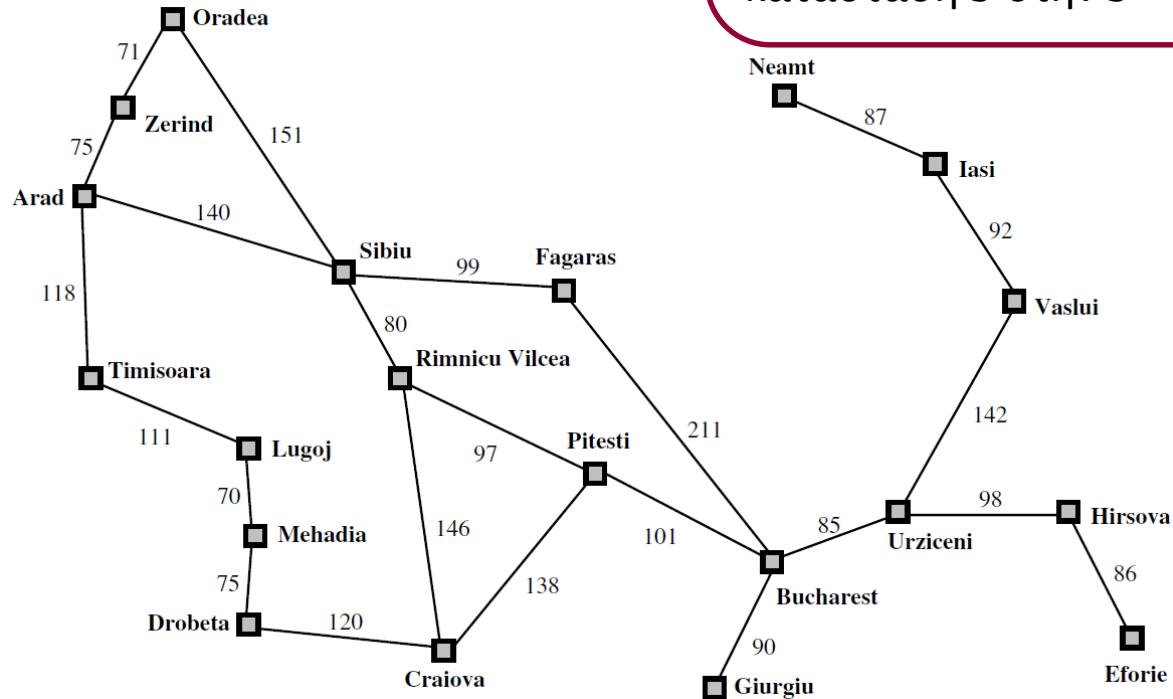
Π.χ.

το συνολικό μήκος της διαδρομής σε χιλιόμετρα

το σύνολο των ενεργειών που απαιτήθηκαν

Υπόθεση:

Το **συνολικό κόστος**
ισούται με το **άθροισμα**
του κόστους των
επιμέρους βημάτων



Κόστος βήματος
 $c(a, s, s')$

Κόστος ενέργειας a για
μετάβαση από την
κατάσταση s στην s'

Καλώς ορισμένα προβλήματα

➤ Λύση

Μια ακολουθία ενεργειών που οδηγούν από την αρχική στην τελική κατάσταση-στόχο.

Αποτίμηση

Μέσω της συνάρτησης κόστους διαδρομής

Βέλτιστη λύση

Η λύση με το **μικρότερο κόστος διαδρομής**
μεταξύ όλων των δυνατών λύσεων

Π.χ. μικρότερη διαδρομή (σε km)

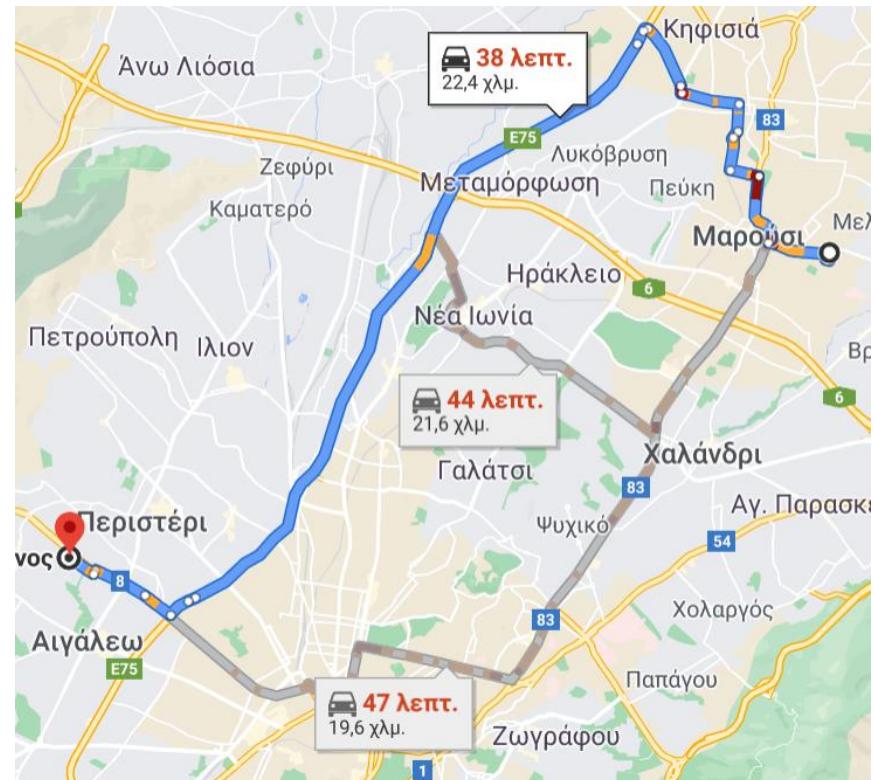
Διατύπωση προβλημάτων

➤ Αφαίρεση (abstraction)

Η διαδικασία **παράλειψης** παραμέτρων/λεπτομερειών που δεν σχετίζονται με την αναπαράσταση του προβλήματος αναζήτησης.

Π.χ.

κατάσταση δρόμου, καιρικές συνθήκες, περιβάλλον, ώρα κλπ



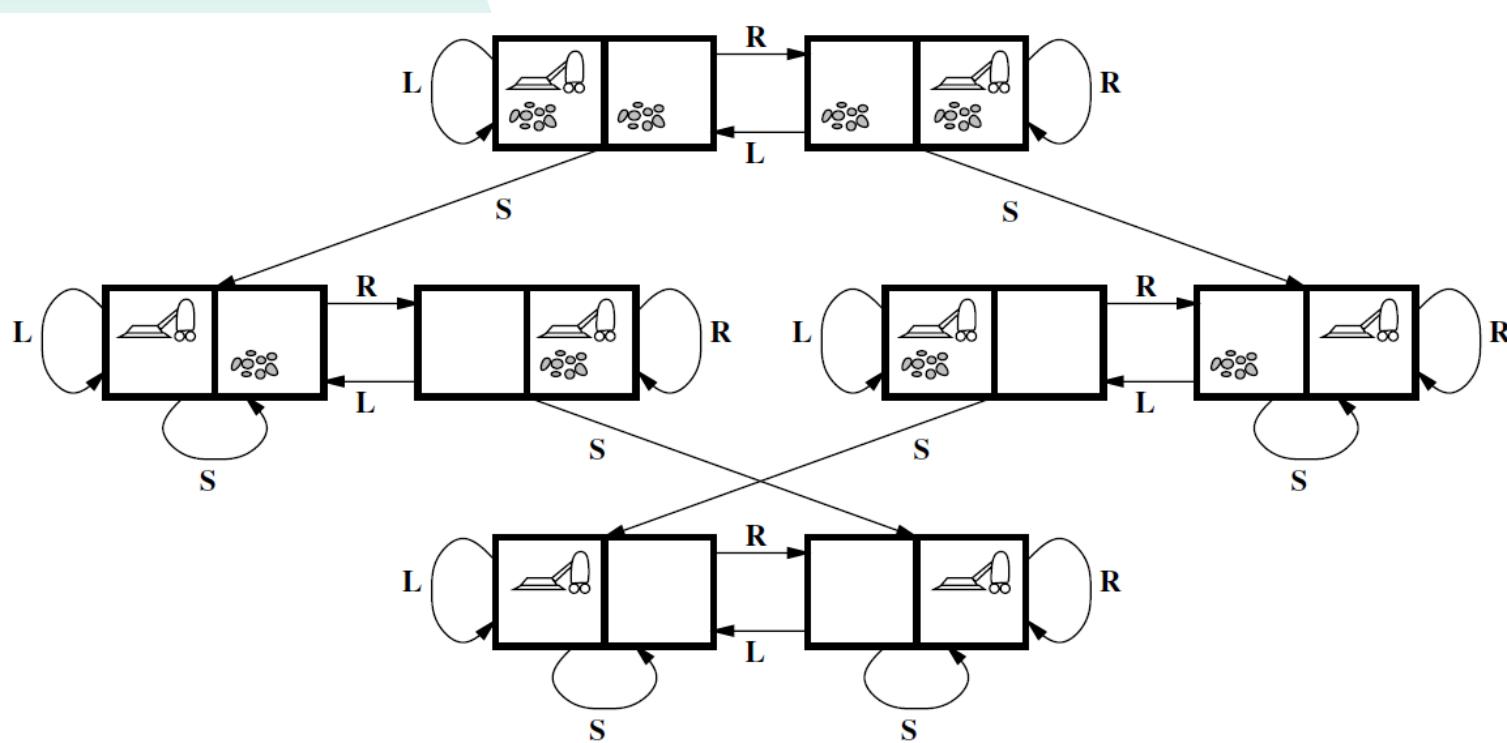
Προβλήματα - παιχνίδια

➤ Γενικά

Σκοπός: η ανάδειξη μεθόδων επίλυσης προβλημάτων.

Χαρακτηριστικά: Επακριβής περιγραφή, χρήση για σύγκριση αποτελεσματικότητας διαφορετικών αλγορίθμων

➤ Πρόβλημα Ηλεκτρικής Σκούπας



Προβλήματα - παιχνίδια

➤ Πρόβλημα Ηλεκτρικής Σκούπας (συν.)

Καταστάσεις

Η σκούπα μπορεί να είναι σε μια από δύο θέσεις.

Κάθε θέση μπορεί να **περιέχει** ή όχι σκόνη.

Υπάρχουν $2 \times 2^2 = 8$ πιθανές **καταστάσεις**

Στην γενική περίπτωση όπου υπάρχουν n θέσεις τότε το πλήθος των πιθανών καταστάσεων είναι $n \times 2^n$

Αρχική κατάσταση

Οποιαδήποτε κατάσταση μπορεί να είναι αρχική.

Ενέργειες

Κάθε κατάσταση έχει τρεις ενέργειες, **L, R, S**

Aριστερά (Left), Δεξιά (Right), Αναρρόφηση (Suck)

Προβλήματα - παιχνίδια

➤ Πρόβλημα Ηλεκτρικής Σκούπας (συν.)

Μοντέλο μετάβασης

Οι τρεις ενέργειες οδηγούν στο αντίστοιχο αποτέλεσμα.

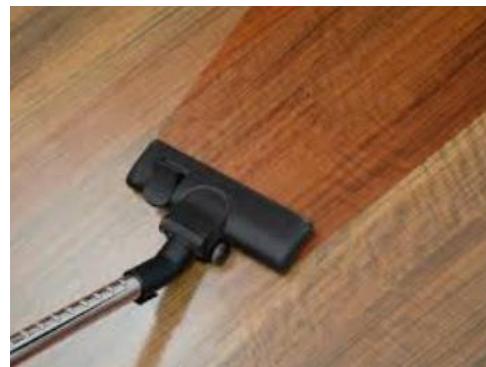
(Εξαίρεση *Left* σε αριστερή θέση, *Right* σε δεξιά θέση και *Suck* σε καθαρή θέση)

Στόχος

Όλοι οι χώροι είναι καθαροί.

Κόστος διαδρομής

Εάν κάθε βήμα έχει κόστος 1 τότε το κόστος της διαδρομής είναι το πλήθος των βημάτων



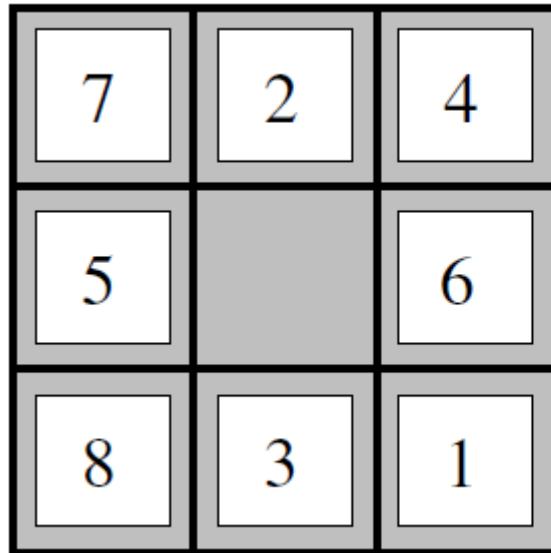
Προβλήματα - παιχνίδια

➤ Παζλ 8 πλακιδίων

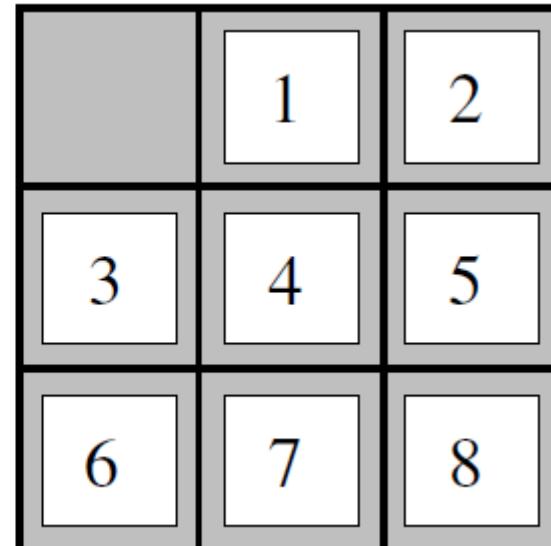
Πίνακας 3x3 με οκτώ **αριθμημένα** πλακίδια και μια **κενή** θέση.

Ένα πλακίδιο που γειτονεύει με την κενή θέση μπορεί να μετακινηθεί σε αυτήν.

Σκοπός είναι να επιτευχθεί μια τελική κατάσταση όπως φαίνεται παρακάτω.



Αρχική κατάσταση



Τελική κατάσταση

Προβλήματα - παιχνίδια

➤ Παζλ 8 πλακιδίων (συν.)

Καταστάσεις

Η κάθε κατάσταση περιγράφεται από την θέση κάθε ενός από τα οκτώ πλακίδια καθώς και της κενής θέσης μέσα στον 3x3 πίνακα

Αρχική κατάσταση

Κάθε κατάσταση μπορεί να είναι αρχική.

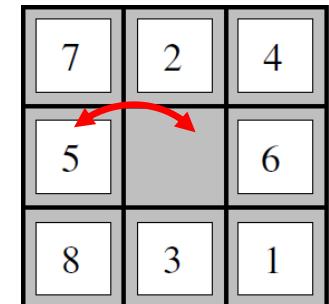
Ενέργειες

Μετακίνηση κάθε πλακιδίου προς

Αριστερά (Left), Δεξιά (Right), Πάνω (Up), Κάτω (Bottom)

Μοντέλο μετάβασης

Δεδομένης μιας κατάστασης και μιας ενέργειας προκύπτει η επόμενη κατάσταση. Π.χ. στην διπλανή κατάσταση, ενέργεια *Αριστερά (Left)* οδηγεί σε αντιμετάθεση του πλακιδίου 5 και της κενής θέσης

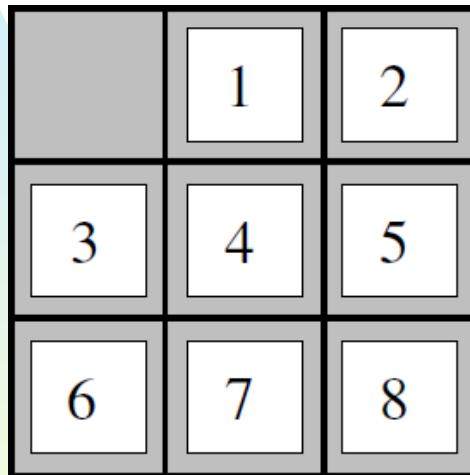


Προβλήματα - παιχνίδια

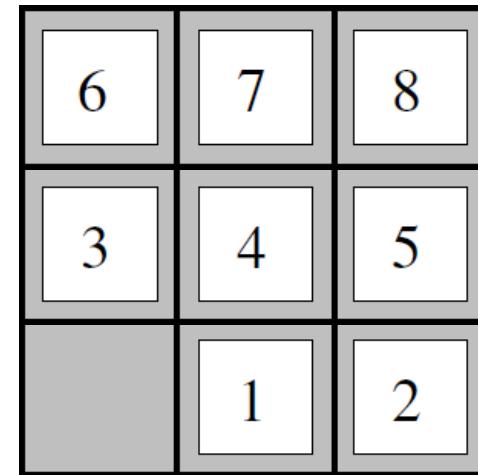
➤ Παζλ 8 πλακιδίων (συν.)

Έλεγχος στόχου

Έλεγχος εάν η κατάσταση ταυτίζεται με την επιθυμητή τελική κατάσταση



Τελική κατάσταση 1



Τελική κατάσταση 2

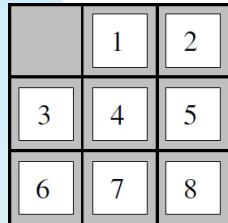
Κόστος διαδρομής

Κάθε βήμα έχει κόστος 1 συνεπώς το κόστος της διαδρομής είναι το **πλήθος των βημάτων**

Προβλήματα - παιχνίδια

➤ Παζλ συρόμενων αντικειμένων

Γενική κατηγορία προβλημάτων



$$\frac{9!}{2} = 181440 \text{ καταστάσεις}$$

| | | | |
|----|----|----|----|
| 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 |
| 9 | 10 | 11 | 12 |
| 13 | 14 | 15 | |

$$\frac{16!}{2} \approx 10^{13} \text{ καταστάσεις}$$

| | | | | |
|----|----|----|----|----|
| 1 | 2 | 3 | 9 | 4 |
| 6 | 14 | 19 | 23 | 5 |
| 16 | 22 | 18 | | 8 |
| 21 | 12 | 7 | 15 | 13 |
| 17 | 11 | 20 | 24 | 10 |

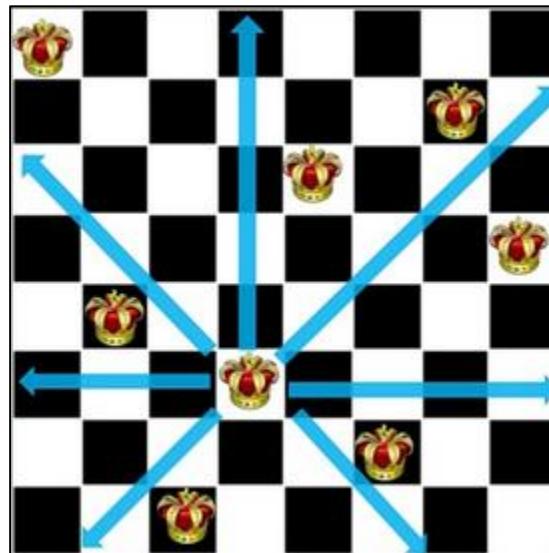
$$\frac{25!}{2} \approx 10^{25} \text{ καταστάσεις}$$

Προβλήματα - παιχνίδια

➤ Πρόβλημα 8 βασιλισσών

Τοποθέτηση οκτώ βασιλισσών σε σκακιέρα έτσι ώστε καμία βασίλισσα να μην απειλείται από άλλη.

(Δηλαδή, για κάθε βασίλισσα να μην υπάρχει άλλη στην ίδια γραμμή, στήλη ή διαγώνιο).



Προβλήματα - παιχνίδια

➤ Πρόβλημα 8 βασιλισσών (συν.)

- 1^η προσέγγιση: Διατύπωση **πλήρους κατάστασης (complete-state)**

Καταστάσεις

Κάθε κατάσταση που περιέχει 0 έως 8 βασίλισσες πάνω στην σκακιέρα.

Αρχική κατάσταση

Κενή σκακιέρα.

Ενέργειες

Προσθήκη βασίλισσας σε κάποια κενή θέση.

Έλεγχος στόχου

Οι 8 βασίλισσες βρίσκονται πάνω στην σκακιέρα και καμία δεν απειλείται.

$$64 \cdot 63 \cdots 57 \approx 1.8 \times 10^{14} \text{ πιθανές ακολουθίες}$$

Προβλήματα - παιχνίδια

➤ Πρόβλημα 8 βασιλισσών (συν.)

- 2^η προσέγγιση: Αυξητική διατύπωση (incremental)

Καταστάσεις

Όλες οι διατάξεις n βασιλισσών, με $0 \leq n \leq 8$, με μία βασίλισσα ανά στήλη στις πρώτες n στήλες από τα αριστερά. Καμία βασίλισσα δεν απειλεί άλλη.

Αρχική κατάσταση

Κενή σκακιέρα.

Ενέργειες

Προσθήκη βασίλισσας σε οποιοδήποτε τετράγωνο της αριστερότερης κενής στήλης ώστε να μην απειλείται από κάποια άλλη βασίλισσα.

Έλεγχος στόχου

Οι 8 βασίλισσες βρίσκονται πάνω στην σκακιέρα και καμία δεν απειλείται.

2057 πιθανές ακολουθίες

Προβλήματα - παιχνίδια

➤ Πρόβλημα 8 βασιλισσών (συν.)

Αυξητική διατύπωση

Εναλλακτική προσέγγιση

Απαγόρευση τοποθέτησης μιας βασίλισσας σε κάποιο τετράγωνο που ήδη απειλείται

Καταστάσεις

Όλες οι διατάξεις n βασιλισσών, με $0 \leq n \leq 8$, με μία βασίλισσα ανά στήλη στις πρώτες n στήλες από τα αριστερά. Καμία βασίλισσα δεν απειλεί άλλη.

Ενέργειες

Προσθήκη βασίλισσας σε οποιοδήποτε τετράγωνο της αριστερότερης κενής στήλης ώστε να μην απειλείται από καμιά άλλη βασίλισσα.

Έλεγχος στόχου

Οι 8 βασίλισσες βρίσκονται πάνω στην σκακιέρα και καμία δεν απειλείται.

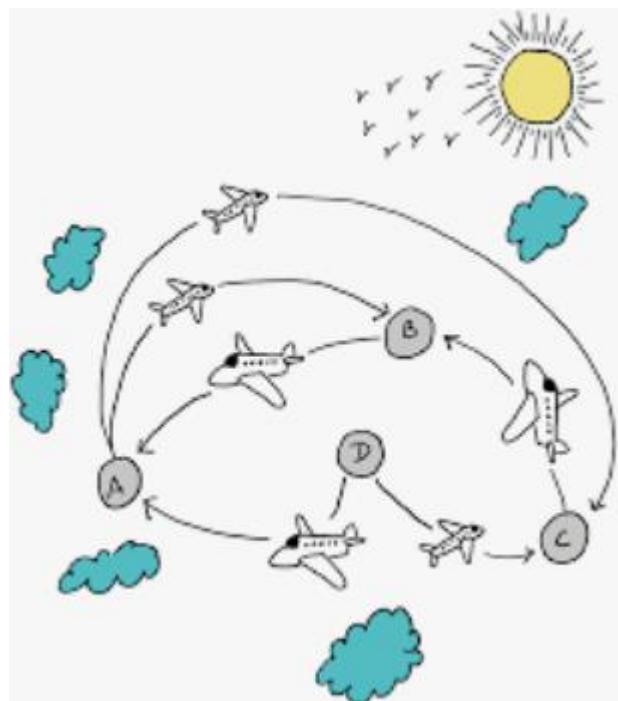
2057 πιθανές καταστάσεις

Προβλήματα πραγματικού κόσμου

➤ Γενικά

Εξετάσαμε ήδη το πρόβλημα **εύρεσης δρομολογίου** (root-finding problem).

- Παρόμοιοι αλγόριθμοι υπάρχουν για την δρομολόγηση σε δίκτυα υπολογιστών, πολεμικές επιχειρήσεις και συστήματα σχεδιασμού αεροπορικών δρομολογίων.
- Βασίζονται συνήθως σε πολύ **σύνθετους περιορισμούς**.



Προβλήματα πραγματικού κόσμου

➤ Προβλήματα περιήγησης

Παρόμοια με εύρεσης δρομολογίου. Διαφορά στην **διατύπωση**:

«Επίσκεψη όλων των πόλεων **τουλάχιστον μια φορά** ξεκινώντας και τερματίζοντας στο Βουκουρέστι»

Ενέργειες

Μετακίνηση μεταξύ γειτονικών πόλεων

Καταστάσεις

Διαφοροποίηση: Η κάθε κατάσταση περιγράφεται από την τρέχουσα πόλη αλλά και όσες έχει ήδη γίνει **επίσκεψη**.

Π.χ. Αρχική κατάσταση

In(Bucharest), Visited({Bucharest})

Τρέχουσα κατάσταση

In(Vaslui), Visited({Bucharest ,Urziceni ,Vaslui})

Έλεγχος στόχου

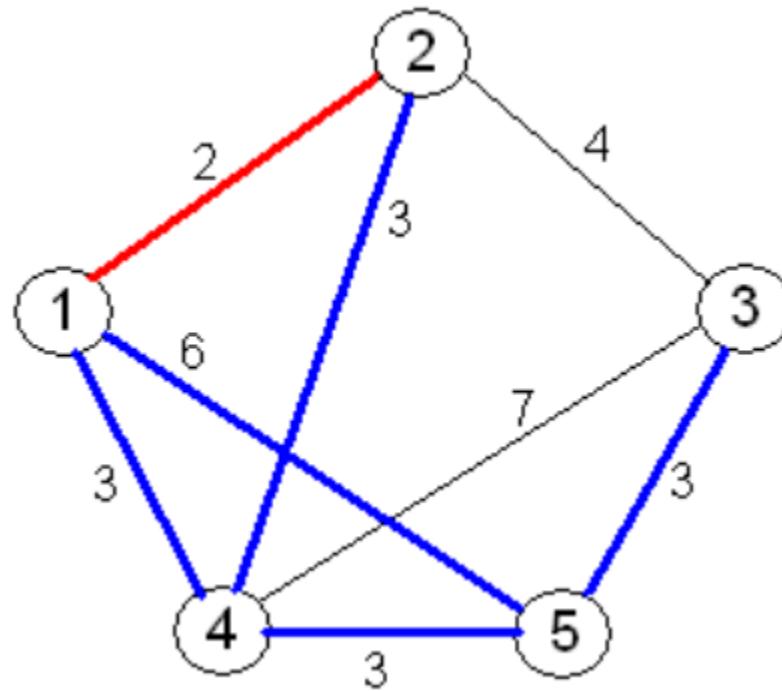
Εάν έχει γίνει επίσκεψη σε όλες τις πόλεις.

Προβλήματα πραγματικού κόσμου

➤ Πρόβλημα πλανόδιου πωλητή

Στο Traveling Salesman Problem, TSP πρέπει να γίνει επίσκεψη σε όλες τις πόλεις ακριβώς μια φορά.

Σκοπός είναι να βρεθεί η **συντομότερη** διαδρομή.



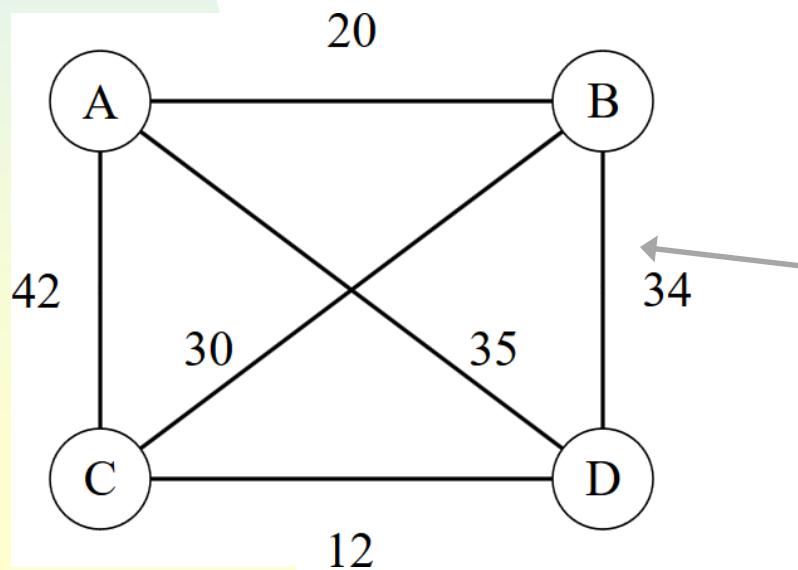
Παραδείγματα

➤ Πρόβλημα πλανόδιου πωλητή (συν.)

Έστω ένα σύνολο από n πόλεις. Κάθε πόλη συνδέεται με κάποια άλλη μέσω ενός δρόμου συγκεκριμένου μήκους.

Βρες την **συντομότερη** διαδρομή που διέρχεται από όλες τις πόλεις

Παραγοντικό $n! = 1 \cdot 2 \cdot \dots \cdot n$



Συνολικά $4! = 1 \cdot 2 \cdot 3 \cdot 4 = 24$
συνδυασμοί (διαδρομές)

ΔΙΑΔΡΟΜΗ ΑΠΟΣΤΑΣΗ

| | |
|------|------|
| ABCD | 62 |
| ABDC | 66 |
| ACBD | 106 |
| ACDB | 88 |
| | |

Στην πραγματικότητα $n/2 = 12$
διαφορετικές διαδρομές
(διότι $ABCD = DCBA$)

Παραδείγματα

➤ Πρόβλημα πλανόδιου πωλητή (συν.)

Έστω ένα σύνολο από n πόλεις. Κάθε πόλη συνδέεται με κάποια άλλη μέσω ενός δρόμου συγκεκριμένου μήκους.

Παραγοντικό $n! = 1 \cdot 2 \cdot \dots \cdot n$

$$5! = 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 120$$

$$6! = 6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 720$$

$$7! = 7 \cdot 6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 5040$$

$$8! = 8 \cdot 7 \cdot 6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 40320$$

$$9! = 9 \cdot 8 \cdot 7 \cdot 6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 362880$$

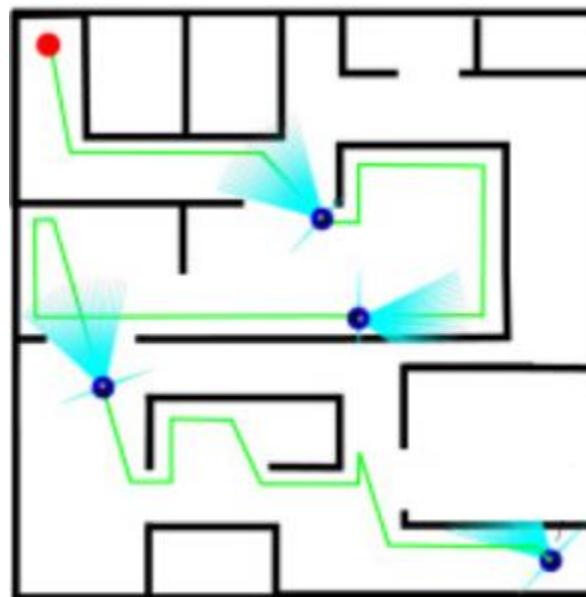
$$10! = 10 \cdot 9 \cdot 8 \cdot 7 \cdot 6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 3628800$$

Προβλήματα πραγματικού κόσμου

➤ Πλοήγηση ρομπότ

Γενίκευση του προβλήματος εύρεσης δρομολογίου

- Μετακίνηση σε **συνεχή** χώρο
- **Απεριόριστο** σύνολο καταστάσεων και ενεργειών
- **Πολυμεταβλητό** πρόβλημα εάν υπάρχει έλεγχος και κινήσεων του ρομπότ
- Εφαρμογή τεχνικών για **περιορισμό** του χώρου καταστάσεων

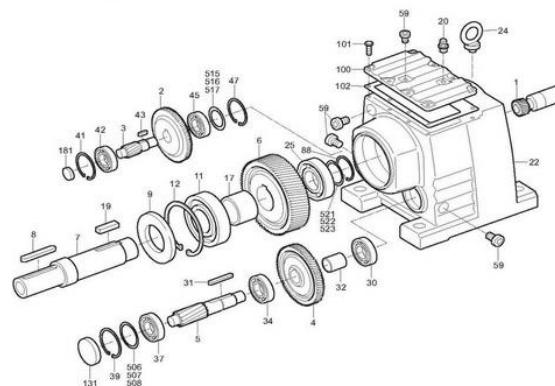


Προβλήματα πραγματικού κόσμου

➤ Αυτόματη ακολουθία συναρμολόγησης

Στόχος: εύρεση σειράς προτεραιότητας με την οποία θα συναρμολογούνται τα μέρη κάποιων αντικειμένων.

- Επιλογή λανθασμένης σειράς δεν επιτρέπει την συνέχιση της ακολουθίας και απαιτεί **αναίρεση** μέρους των βημάτων που έχουν ήδη γίνει
 - Ο έλεγχος της σκοπιμότητας ενός βήματος της ακολουθίας είναι ένα δύσκολο γεωμετρικό πρόβλημα που σχετίζεται με την πλοϊγηση ρομπότ.
 - Η παραγωγή των επόμενων **επιτρεπτών ενεργειών** είναι το πιο δαπανηρό μέρος της ακολουθίας συναρμολόγησης.
 - Οποιοσδήποτε πρακτικός αλγόριθμος θα πρέπει να **αποφεύγει την διερεύνηση όλου του χώρου καταστάσεων** και να περιορίζεται σε ένα μικρό μέρος του.



Αναζήτηση λύσεων

➤ Αλγόριθμοι αναζήτησης

Λύση σε ένα πρόβλημα είναι μια ακολουθία ενεργειών.

Συνεπώς, οι αλγόριθμοι αναζήτησης διερευνούν διάφορες **πιθανές ακολουθίες ενεργειών**.

➤ Δένδρο αναζήτησης

Απαρτίζεται από πιθανές ακολουθίες ενεργειών.

Στην **ρίζα** βρίσκεται η αρχική κατάσταση.

Οι **κλάδοι** αντιστοιχούν σε ενέργειες και οι **κόμβοι** αντιστοιχούν σε καταστάσεις στον χώρο καταστάσεων του προβλήματος.

Απαιτείται να γίνεται **έλεγχος** εάν η τρέχουσα κατάσταση είναι η τελική κατάσταση-στόχος.

Εάν δεν βρεθεί ο στόχος εξετάζονται άλλες καταστάσεις μέσω της **επέκτασης** (expanding) της τρέχουσας κατάστασης.

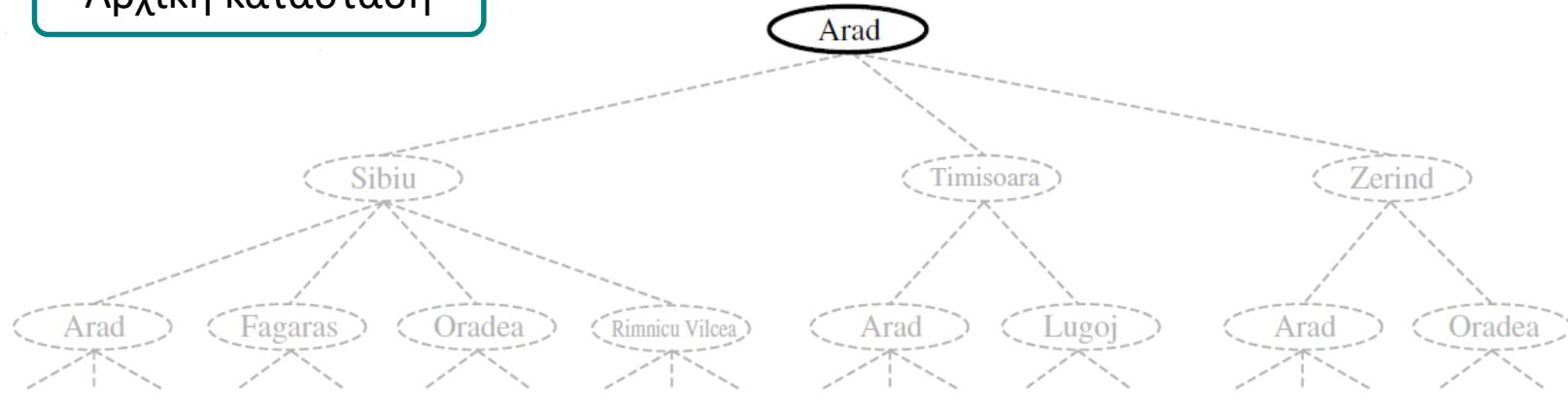
Εφαρμογή της **συνάρτησης διαδόχων** για την παραγωγή νέου συνόλου καταστάσεων.

Αναζήτηση λύσεων

➤ Παράδειγμα

Δένδρο μερικής αναζήτησης για εύρεση της διαδρομής από Arad προς Βουκουρέστι.

Αρχική κατάσταση



Αρχική κατάσταση στην οποία ξεκινά η επίλυση του προβλήματος.

$In(Arad)$

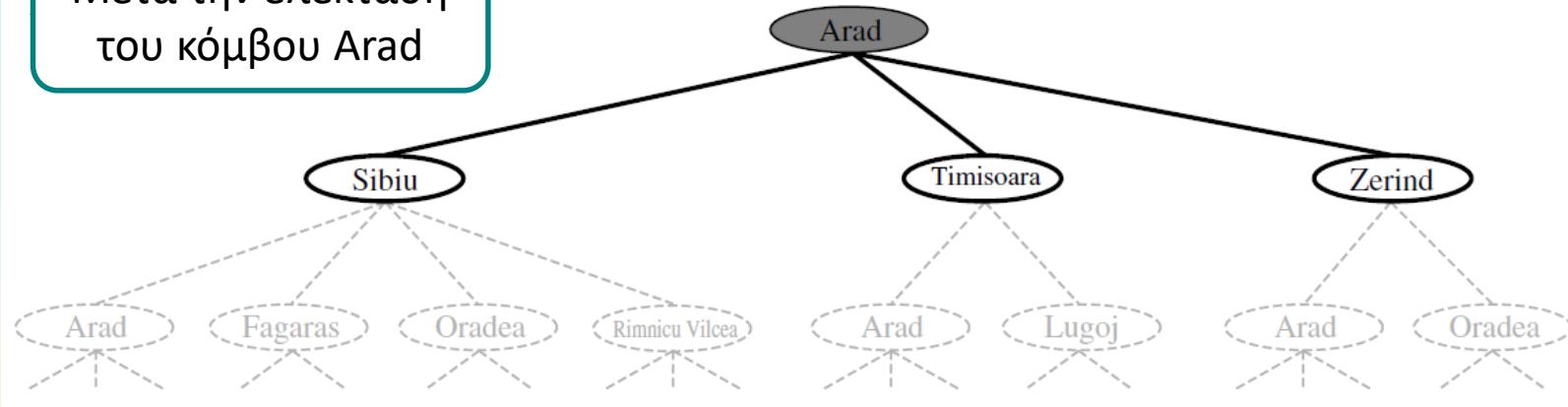
Έλεγχος εάν είναι η τελική κατάσταση-στόχος.

Αναζήτηση λύσεων

➤ Παράδειγμα (συν.)

Επέκταση του δένδρου και δημιουργία νέων κόμβων.

Μετά την επέκταση
του κόμβου Arad



Τρεις νέες καταστάσεις

$In(Sibiu)$, $In(Timisoara)$, και $In(Zerind)$.

Μέτωπο αναζήτησης είναι το διατεταγμένο σύνολο των κόμβων που δεν έχουν ακόμη επεκταθεί προς διάδοχους κόμβους, δηλαδή το σύνολο των διαθέσιμων κόμβων.

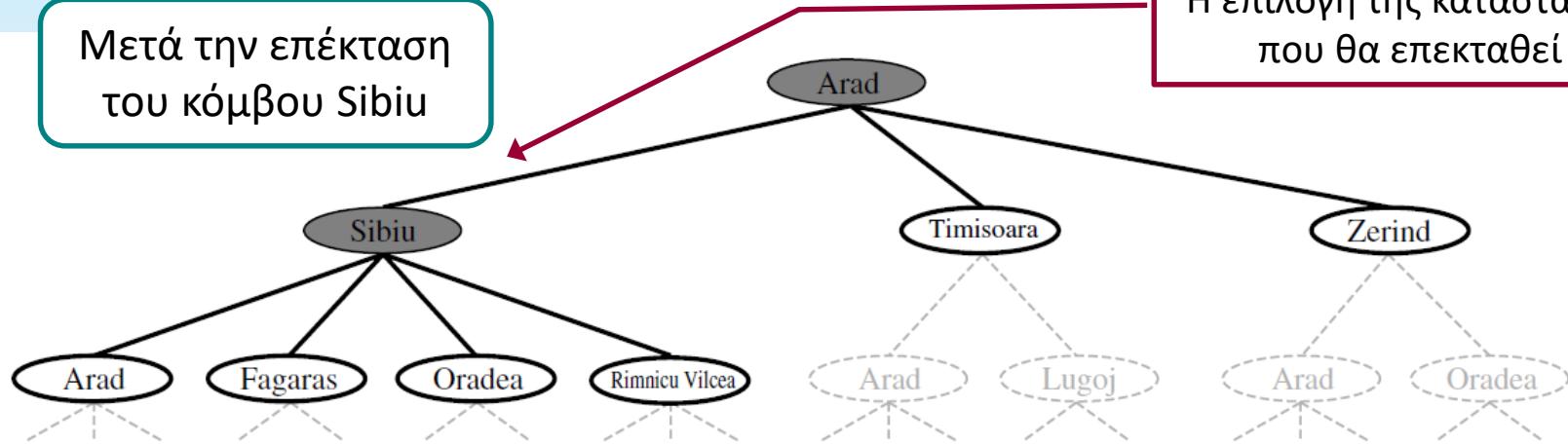
Αναζήτηση λύσεων

➤ Παράδειγμα (συν.)

Επιλογή κόμβου Sibiu μεταξύ των διαθέσιμων.

Μετά την επέκταση
του κόμβου Sibiu

Στρατηγική αναζήτησης
Η επιλογή της κατάστασης
που θα επεκταθεί



Επέκταση κόμβου

Τέσσερις νέες καταστάσεις

$In(Arad), In(Fagaras), In(Oradea), \text{ and } In(RimnicuVilcea)$

Σύνολο διαθέσιμων καταστάσεων: 6

Αναζήτηση λύσεων

➤ Άτυπος (γενικός) αλγόριθμος αναζήτησης

```
function Tree – Search (πρόβλημα) returns μια λύση ή αποτυχία
αρχικοποίηση του δένδρου (από αρχική κατάσταση προβλήματος)

loop do
    if δεν υπάρχουν υποψήφιοι για επέκταση then
        return αποτυχία
    επιλογή ενός κόμβου-φύλλου προς επέκταση, σύμφωνα με τη στρατηγική
    if ο κόμβος περιέχει μια κατάσταση-στόχου then
        return την αντίστοιχη λύση
    else
        επέκταση κόμβου και προσθήκη των κόμβων που προκύπτουν στο δένδρο
        αναζήτησης
```

Αναζήτηση λύσεων

➤ Επαναλαμβανόμενες καταστάσεις

Στο παράδειγμα υπάρχουν 20 καταστάσεις (πόλεις)

Η κατάσταση $In(Arad)$ είναι **επαναλαμβανόμενη** λόγω της διαδρομής

$$Arad \rightarrow Sibiu \rightarrow Arad$$

- Το πλήρες δέντρο αναζήτησης είναι άπειρο (δεν υπάρχει όριο στο πόσες φορές θα επαναληφθεί ο βρόγχος)
- Χρήσιμη ιδιότητα: Το κόστος ενός βρόγχου είναι πάντα μεγαλύτερο από την ίδια διαδρομή χωρίς τον βρόχο (προϋπόθεση: προσθετικά και μη-αρνητικά κόστη)

Αναζήτηση λύσεων

➤ Επαναλαμβανόμενες διαδρομές

Συμβαίνουν όταν η μετακίνηση από ένα κόμβο σε άλλον μπορεί να γίνει με περισσότερους από έναν τρόπους

Arad → Sibiu (140 km)

Arad → Zerind → Oradea → Sibiu (297 km)

- Η δεύτερη διαδρομή είναι **περιττή** (redundant path)

Είναι αναπόφευκτες σε πολλά προβλήματα αναζήτησης (π.χ. προβλήματα εύρεσης δρομολογίου, παζλ συρόμενων αντικειμένων)

Λύση

Χρήση συνόλου ελεγμένων κόμβων

- Περιέχει τους κόμβους που έχουν **ήδη** εξετασθεί
- Ένας καινούργιος κόμβος προστίθεται στο σύνολο μόνο εάν **δεν** υπάρχει ήδη σε αυτό

Αναζήτηση λύσεων

- Άτυπος (γενικός) αλγόριθμος αναζήτησης (με έλεγχο επαναλήψεων)

function *Graph – Search* (*πρόβλημα*) **returns** μια λύση ή αποτυχία αρχικοποίηση του δένδρου (από αρχική κατάσταση προβλήματος)

αρχικοποίηση κενού συνόλου ελεγμένων κόμβων

loop do

if δεν υπάρχουν υποψήφιοι για επέκταση **then**

return αποτυχία

επιλογή ενός κόμβου-φύλλου προς επέκταση, σύμφωνα με τη στρατηγική

if ο κόμβος περιέχει μια κατάσταση-στόχου **then**

return την αντίστοιχη λύση

else

προσθήκη κόμβου στο σύνολο ελεγμένων κόμβων

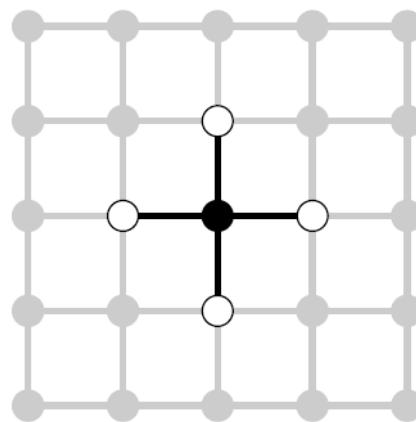
επέκταση κόμβου και προσθήκη των κόμβων που προκύπτουν στο δένδρο

αναζήτησης **μόνο εάν ΔΕΝ υπάρχουν ήδη στο σύνολο ελεγμένων κόμβων**

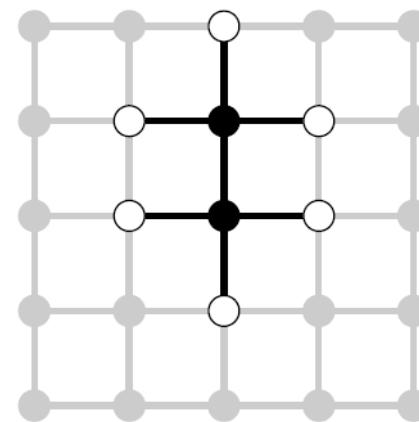
Αναζήτηση λύσεων

➤ Παράδειγμα 1

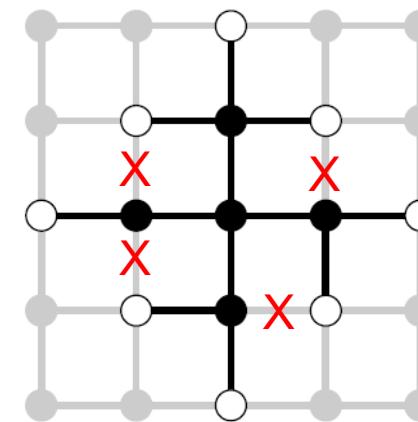
Πρόβλημα εύρεσης δρομολογίου σε **ορθογώνιο πλέγμα** (ιδιαίτερα σημαντικό σε παιχνίδια υπολογιστών)



(a)



(b)



(c)



κόμβοι επέκτασης
ελεγμένοι κόμβοι
μη ελεγμένοι κόμβοι

- (a) επέκταση του αρχικού κόμβου
- (b) επέκταση ενός κόμβου-φύλλου
- (c) επέκταση των υπόλοιπων κόμβων-φύλλων
(δεξιόστροφα)

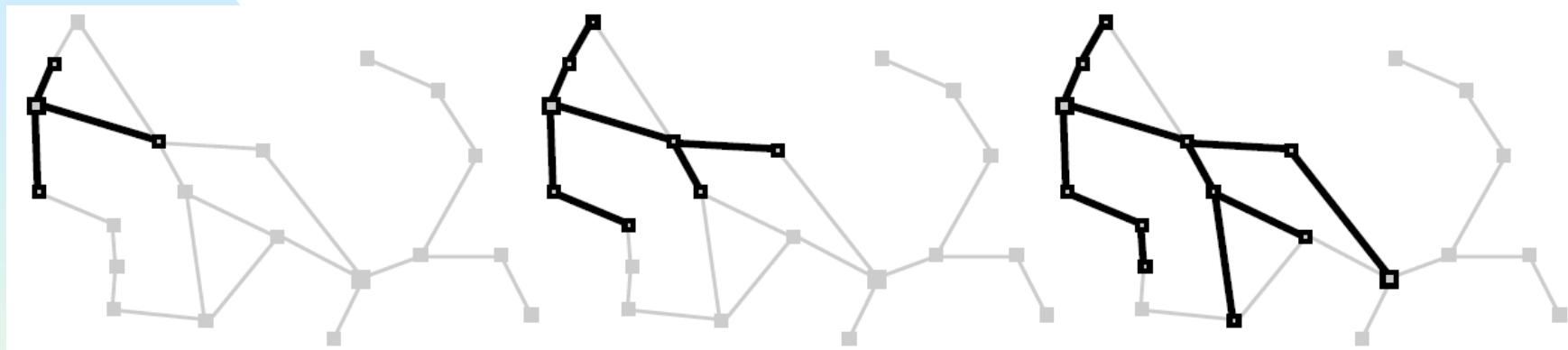
Κόμβοι που έχουν ήδη ελεγχθεί **παραλείπονται**

Κάθε κόμβος υπάρχει το πολύ **μια φορά** μέσα στο δέντρο αναζήτησης

Αναζήτηση λύσεων

➤ Παράδειγμα 2

Πρόβλημα εύρεσης δρομολογίου σε χάρτη



Ακολουθία επεκτάσεων σε αναζήτηση σε γράφο (πρόβλημα Ρουμανίας)

- Σε κάθε επίπεδο ο γράφος επεκτάθηκε κατά ένα βήμα
- Στο τελευταίο επίπεδο η πιο βόρεια πόλη έγινε αδιέξοδο διότι και οι δύο πόλεις που γειτονεύει έχουν ήδη ελεγχθεί από άλλες διαδρομές

Ιδιότητες *Graph – Search*

- Διαχωρισμός των κόμβων (καταστάσεις) σε ελεγμένους και μη
- Η διαδρομή από την αρχή προς κάθε καινούργιο κόμβο πρέπει να περάσει από κάποιον ελεγμένο κόμβο