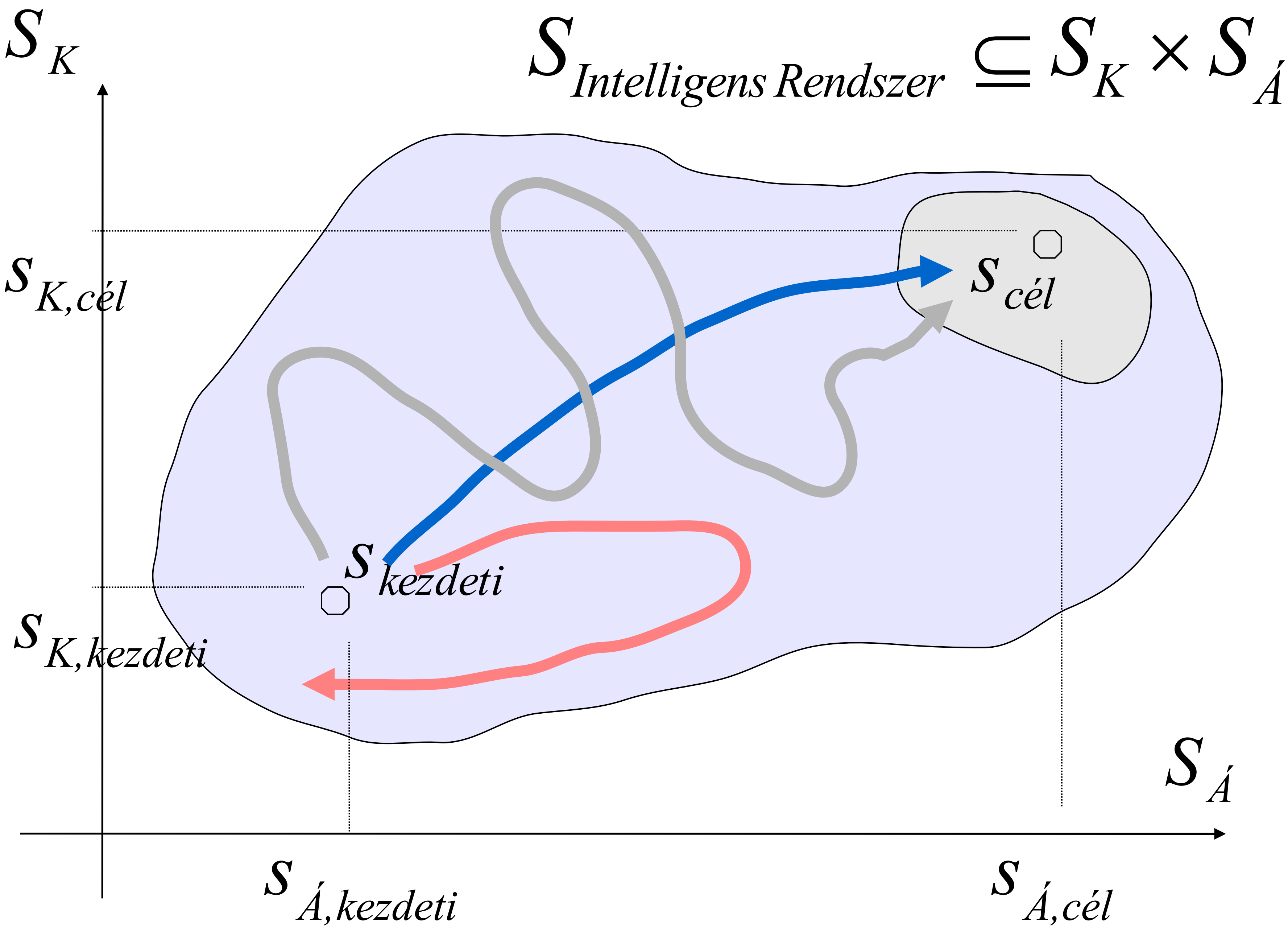
Mesterséges Intelligencia MI

Problémamegoldás kereséssel – vakon

http://mialmanach.mit.bme.hu/aima/ch03s03
3. fejezet 3.4 alfejezet

Pataki Béla,
(Hullám Gábor)

BME I.E. 414, 463-26-79
pataki@mit.bme.hu,
http://www.mit.bme.hu/general/staff/pataki



S_K
$S_{Intelligens\ Rendszer} \subseteq S_K \times S_Á$
$s_{K,cél}$
$s_{cél}$
$s_{kezdeti}$
$s_{K,kezdeti}$
$S_Á$
$s_{Á,kezdeti}$
$s_{Á,cél}$

Keressük meg azt a jó trajektóriát (utat), ami a kezdeti állapotból a célállapotba visz, a lehető legkisebb költséggel!

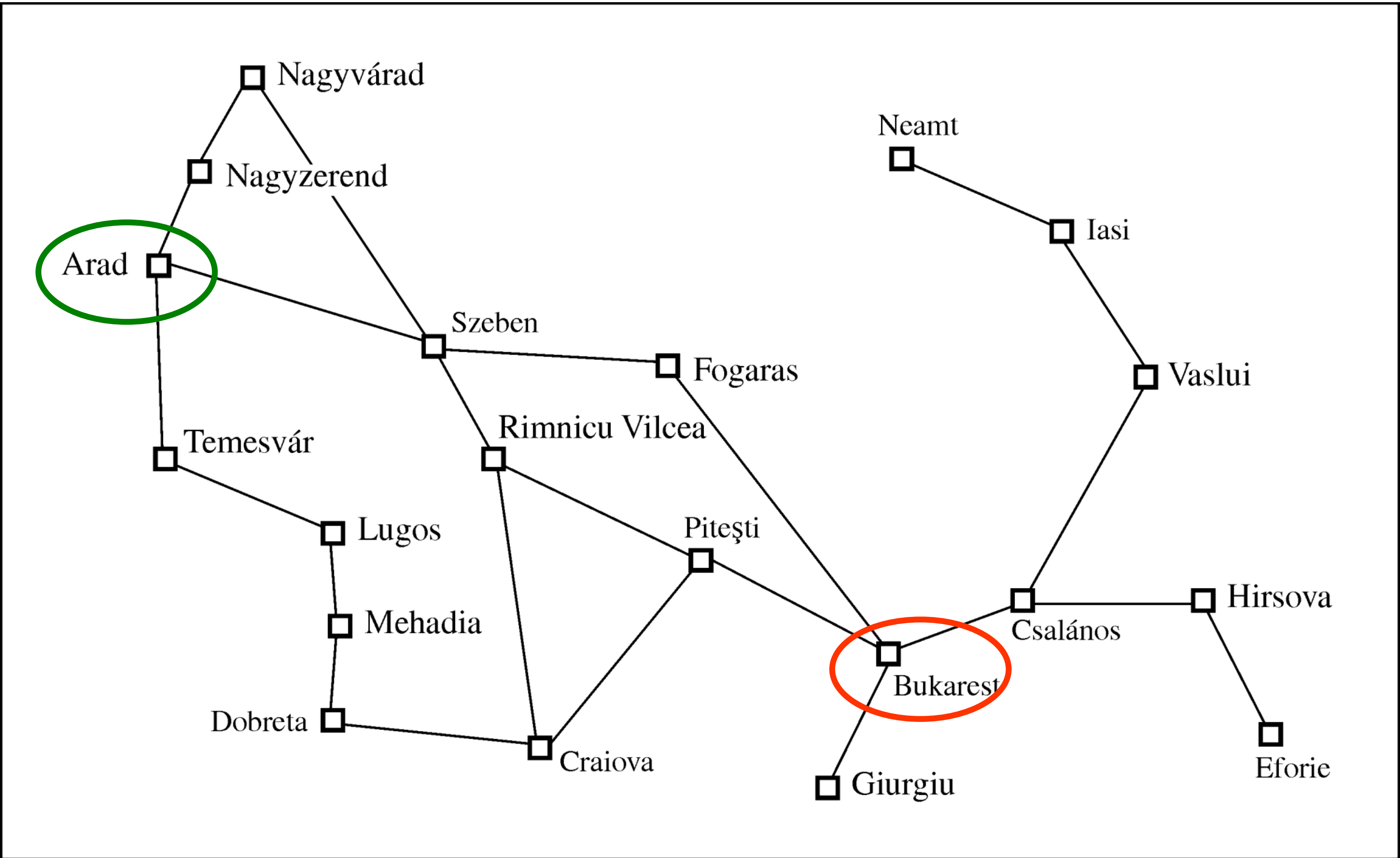
A feladat könnyűnek tűnik, de ha nagyon sok állapot van, sokféle átmenet lehet egyikből a másikba, és nem látjuk át „felülről” az állapotteret, akkor bajba kerülhetünk.

Melyik a jó keresési eljárás?

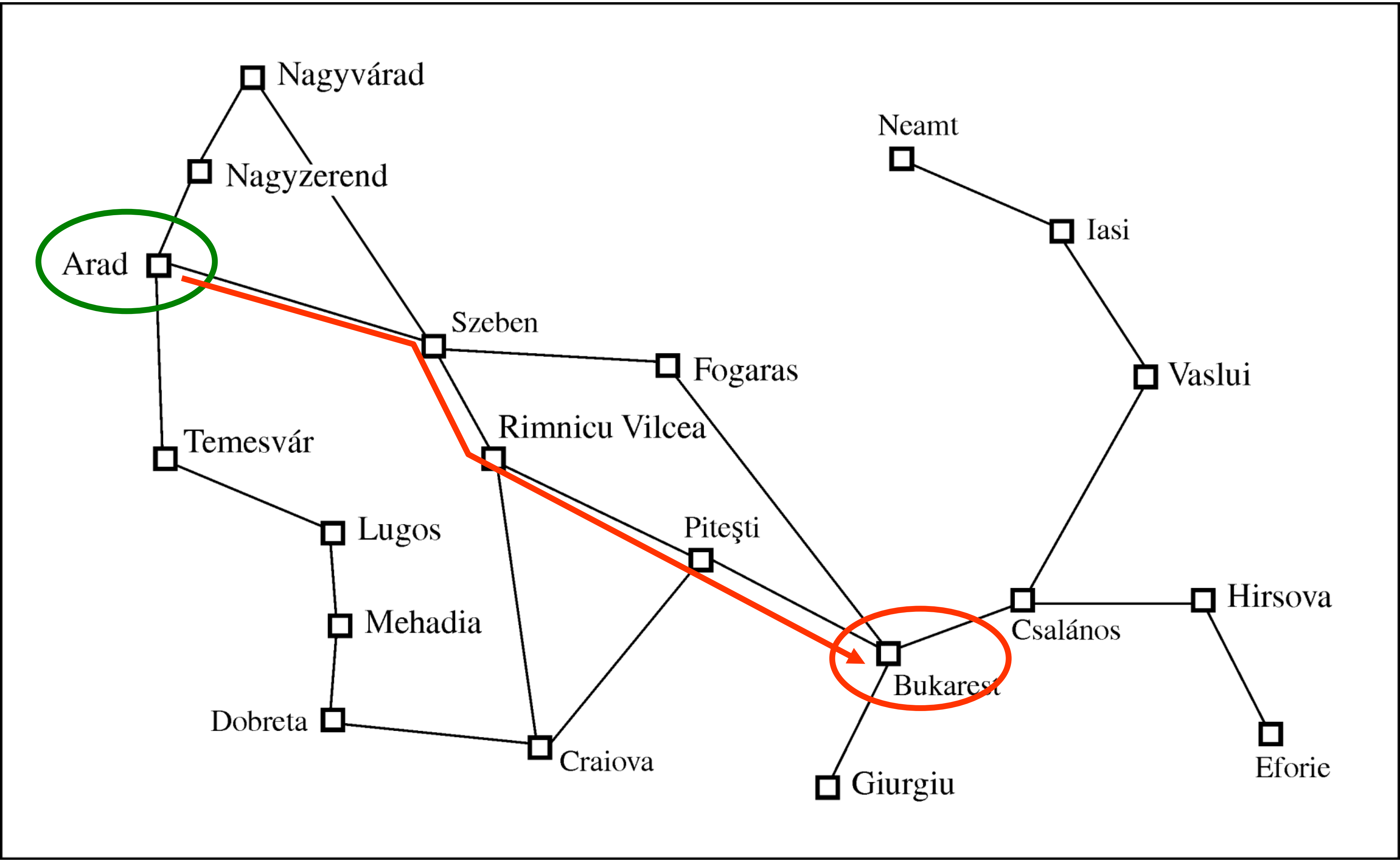
Részben az elért eredmény, részben annak megtalálási költsége minősíti. A szokásos értékelési szempontok:

1. Teljesség (*completeness*) - ha van megoldás, biztosan megtalálja
2. Időigény (*time complexity*) – mennyi idő a megoldás megtalálása?
3. Tárigény (*space complexity*) – mennyi tárhely kell
4. Optimalitás (*optimality*) – ha több megoldás van, megtaláljuk-e a legjobbat? (mi a legjobb? – sokszor izgalmas kérdés!)

A Russel-Norvig könyv példája: el akarunk jutni Aradról Bukarestbe...



Milyen utat találunk meg? Például ezt?



Keresési stratégiák – 1

Nem informált keresések (un. **gyenge**, vagy **vak** keresések)

a. <u>tudjuk</u>: hogy néz ki a célállapot

b. <u>egyáltalán nincs infónk arról</u>: milyen költségű az aktuálisból a célállapotba vezető út

A vak (nem informált) keresési stratégiákat a **csomópontok kifejtési sorrendje különbözteti meg**. Ez a különbség óriási jelentőséggel bírhat.

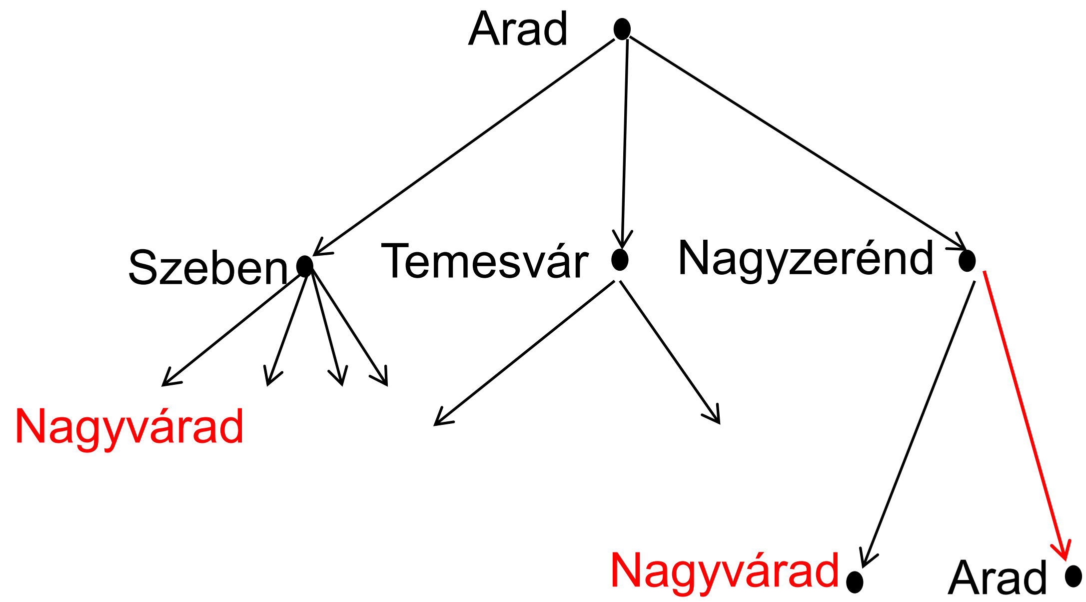
Múlt óra végén:

1. Szélességi keresés (width first search)
2. Egyenletes költségű keresés (uniform cost search $\subset$ best first search)
3. Mélységi keresés (depth first search)

A keresés folyamatát az ún. **keresési fával** ragadjuk meg.

A gráf csomópontjainak adatszerkezete (pl.):

1. az állapottérnek a csomóponthoz tartozó állapota
2. szülő csomópont
3. a csomópontot generáló operátor
4. a gyökértől eddig a csomópontig vezető út csomópontjainak száma (mélység – *depth*)
5. (a csomópontig tartó útköltség)



Egyetlen „Nagyvárad” állapot van, de több csomóponthoz is tartozhat ez az állapot.

Fontos fogalmak:

- kifejtés (*expand*)
- (átl.) elágazási tényező – *b* (*branching factor*)
- mélység – *d* (*depth*)
- hullámfront (*frontier*)

Mélységi keresés

rohanás előre, egy ág mentén, de a visszalépés lehetőségével (pl. sakk: d2-d4-el nyitunk, és először kizárólag a sötét d7-d5 válaszát bontjuk ki, annak is egy alváltozatát, al-alváltozatát stb.)

Mélységkorlátozott keresés

az utak maximális mélységére egy vágási korlátot ad.

Románia jelen egyszerűsített térképe: 20 város. Ha létezik egy megoldás, az maximálisan 19 lépés hosszú lehet.
Minden város bármelyik városból legfeljebb 9 lépésben elérhető:
az állapottér **átmérője** = jobb mélységkorlát.
A mélységi korlát elérésénél a keresés visszalép. A megoldást, amennyiben létezik és a mélységkorlátnál sekélyebben fekszik, garantáltan megtaláljuk.
De <u>semmi garancia</u> nincs arra, hogy a <u>legkisebb költségű</u> (itt: legrövidebb út) megoldást találjuk meg.
Amennyiben túl kis mélységkorlátot választunk, akkor a mélységkorlátozott keresés még csak <u>teljes sem lesz</u>.

A híd (logikai feladvány)
Egy ködös este 4 hallgató megrekedt a Borzalmas Titkok Szigetén, ahonnan a szárazföldre csak egy rozoga híd vezetett át. Az éjszakát a szigeten életveszélyes volna eltölteni, a rothadó híd egyszerre csak 2 személyt bír el és sötétben nem járható. Viszont a közelgő sötétséghez csak egy zseblámpájuk volt, melynek eleme még 17 perc világításra volt elég.

Dani, a maratonfutó, 1 perc alatt menne át a túlsó oldalra. Cili, a biciklista, 2 perc alatt végig tudná futni a hidat. A jó tornász Bercel 5 perc alatt kocogná végig. Viszont Andrásnak, az erős dohányosnak, 10 percre lenne szüksége.

Sikerül-e mindannyiuknak megmenekülni a szigetről? Ha nem, miért nem, ha igen, hogyan?

Megoldás: a táblán (mélységkorlátozott kereséssel)

Iteratívan mélyülő keresés

Pár percen belül kvíz!!!

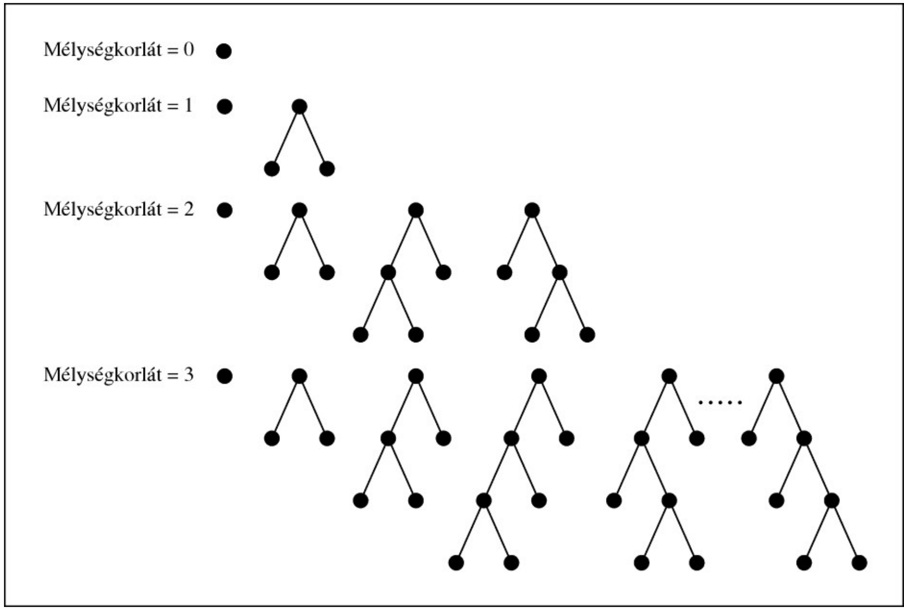
Slate és Atkin (1977): CHESS 4.5 sakkprogram.

- mélységkorlátozott keresés - egy jó mélységkorlát megválasztása?

A legtöbb esetben azonban mindaddig nem tudunk jó mélységkorlátot adni, amíg meg nem oldottuk a problémát.

A legjobb mélységkorlát mechanikus kiválasztása:

- kipróbálja az összes lehetséges mélységkorlátot: először 0, majd 1, majd 2, stb.
- mélységkorláttal végez mélységkorlátozott keresést.



Kvíz

Iteratívan mélyülő keresést végzünk, és a keresési fában minden csomópontunknak 4 gyermek-csomópontja van. Ha 10 mélységben találjuk meg a megoldást (a lehető leghosszabb lépésszám után), akkor hányszor több csomópontkifejtést végeztünk, mint ha egyszerű mélységkorlátozott keresést végeztünk volna 10 mélységkorláttal?

A. Nagyságrendileg másfélszer több csomópontkifejtést
B. Nagyságrendileg tízszer több csomópontkifejtést
C. Nagyságrendileg ezerszer több csomópontkifejtést
D. Nagyságrendileg tízezerszer több csomópontkifejtést

Iteratívan mélyülő keresés gyakorlatilag ötvözi a szélességi és mélységi keresés előnyös tulajdonságait

A szélességi kereséshez hasonlóan **optimális** és **teljes**, de csak a mélységi keresés **szerény** memória igényével rendelkezik.

De bizonyos állapotokat az algoritmus többször is kifejt!

Tékozló? - egy exponenciális keresési fában majdnem az összes csomópont a legmélyebb szinten található

d mélységben a megoldás, *b* elágazási tényező: szélességi/mélységi keresésnél a kifejtések száma: $1 + b + b^2 + ... + b^{d-2} + b^{d-1} + b^d$

pl. $b = 10$ és $d = 5$ esetén ez a szám: $1 + 10 + 100 + ... + 100000 =$ **111.111**

az iteratívan mélyülő keresésnél a kifejtések teljes száma:

$(d+1)1 + (d)b + (d-1)b^2 + ... + 3b^{d-2} + 2b^{d-1} + 1b^d$

$b = 10$ és $d = 5$ esetén: $6 \cdot 1 + 5 \cdot 10 + 4 \cdot 100 + ... +$ **$1 \cdot$ 100.000** = **123.456**
(csak +23,5%)

minél nagyobb az elágazási tényező, annál kisebb a többletmunka
(pl. b = 2 elég rossz, de ott is csak kb. **200%**, kb. *kétszerese* a szélességi/mélységinek!)

Kétirányú keresés

- egyszerre előrefelé a kiinduló állapotból, illetve hátrafelé a cél állapotból
- a keresés akkor fejeződik be, ha a két keresés valahol találkozik

$$O(2 \times b^{d/2}) = O(b^{d/2})$$

Például: $b = 10$, $d = 6$:

a szélességi/mélységi keresés = **1.111.111** csomópont,

a kétirányú keresés mindkét irányban 3 mélységnél ér célba = **2.222** csomópontot generál.

Elméletben nagyon jó, az implementálás nem triviális.

Célállapotból hátrafelé keresni? Az *n* csomópont **elől csomópontjai** azon csomópontok, amelyek követő csomópontjalehet *n*.

- A hátrafelé keresés a cél csomópontból indulva az elől csomópontok egymást követő generálását jelenti. Megfordítható-e a folyamat? (Összerakható-e Csernobil a romokból?)
- Pl. sakk, nagyon sok célállapot, melyikből induljunk visszafele?

Ha az összes operátor reverzibilis, akkor az előd és követő halmazok azonosak. Néhány probléma esetén azonban az elődök meghatározása nagyon nehéz.

Mi van, ha nagyon sok cél állapot létezik?

a cél állapotok egy ***explicit*** listája

a cél állapotok egy ***leírása***

Például a sakkban mik a sakk-matt célállapotok (nem egy célállapot van!) megelőző állapotai?

Hatékony módszer kell arra, hogy: egy frissen generált csomópont megjelenik-e már a másik fél keresési fájában.

El kell tudni dönteni, hogy az egyes félrészekben milyen keresésre fog sor kerülni.

$O(b^{d/2})$ komplexitás feltételezi, hogy a két hullámfront metszésének megállapítása konstans idő alatt elvégezhető(?)

Hogy a két keresés találkozzon, legalább az egyik keresés összes csomópontját memóriában kell tartani = tár $O(b^{d/2})$.

A neminformált keresési stratégiák összehasonlítása

b - elágazási tényező,
m - a keresési fa maximális mélysége,
d - a megoldás mélysége,
l - a mélység korlát.

Jellem-ző	SzK	EgyKK	MK	MKK	IMK	KK
Idő-igény	b^d	b^d	b^m	b^l	b^d	$b^{d/2}$
Tár-igény	b^d	b^d	bm	bl	bd	$b^{d/2}$
Opt.?	Igen (ha…)	Igen	Nem	Nem	Igen	Igen
Teljes?	Igen	Igen	Nem	Igen, ha $l \geq d$	Igen	Igen

A neminformált keresési stratégiák összehasonlítása

b - elágazási tényező, *d* - a megoldás mélysége, *m* - a keresési fa maximális mélysége,
l - a mélység korlát.

Krit.	SzK	EgyKK	MK	MKK	IMK	KK
Idő igény	b^d	b^d	b^m	b^l	b^d	$b^{d/2}$
Tár igény	b^d	b^d	bm	bl	bd	$b^{d/2}$
Opt.?	Igen	Igen	Nem	Nem	Igen	Igen
Teljes?	Igen	Igen	Nem	Igen, ha $l \geq d$	Igen	Igen

Az exponenciális tár-, illetve időigény komoly problémát jelent!
Jobb módszerek kellenek!