Tekoälyn (58066-7) III laskuharjoitustehtävät (viikolle 42/2001)

Tehtävä III.1: Eräästä pasianssikirjasta löytyy mm.

> **KIRPPUPASIANSSI**
>
> Samoin kuin hauskassa lastenpelissä, kirppupelissä, nappulat hyppivät toistensa yli, tässä pasianssissa kortit hyppivät toistensa yli. Pasianssilla on myös nimi *sukkanauha*, koska sen asetelma kuminauhan tavoin venyy ja supistuu. Pasianssi on helppo ja hauska, mutta se ei merkitse sitä, että se menisi liian helposti läpi.
>
> **Kortit**
> Tavallinen 52 kortin pakka.
>
> **Asetelma ja pelin kulku**
> Jakakaa kortteja vaakasuoraan riviin. Kun kaksi samaa maata olevaa korttia tai kaksi samanarvoista korttia sattuu vierekkäin, voitte nostaa oikeanpuoleisen vasemmanpuoleisen päälle.
>
> Kortti voi hypätä vasemmalta oikealle myös kahden sopimattoman kortin yli toisen samaa maata olevan tai samanarvoisen kortin päälle.
>
> Asetelmaan muodostuu vähitellen korttipinoja, ja samat hyppysäännöt ovat voimassa niidenkin suhteen edellyttäen, että pinon päällimmäinen kortti on hyppyyn oikeutettu.
>
> Muistakaa, että kortti saa hypätä vain viereisen kortin päälle tai kahden kortin yli – ei siis koskaan yhden kortin yli eikä myöskään kolmen kortin yli.
>
> Pasianssi on mennyt läpi, kun kaikki kortit ovat kerääntyneet yhdeksi ainoaksi suureksi pinoksi – mutta siten käy melko harvoin. Tyydyttävänä tuloksena voi pitää korttien kerääntymistä jo neljäänkin eri maita olevaan ja eriarvoiseen pinoon.

Intuitiivisesti vastaava ongelma on seuraava: ”Tässä on sekoitettu korttipakka. Selaile sitä ja päättele, voisiko kirppupasianssin pelata sillä läpi. Ja jos voisi, niin miten?”

Muotoile se formaalisti luentojen mukaisena etsintäongelmana.

Tehtävä III.2: Valitse luennoista sopiva etsintäalgoritmi tehtävän III.1 muotoilullesi. Ota huomioon seuraavat tekijät:

- Onko toistuvien tilojen poistaminen tarpeen? Järkevää?
- Onko algoritmisi täydellinen? Aina vai juuri tähän ongelmaan?
- Onko tarvittava muistin määrä kohtuullinen? Tai voidaanko sitä säästää?

Tehtävä III.3: Joskus on vaikeaa keksiä sellaista heuristista funktiota h, joka kertoisi annetun solmun n ”laadun” yhtenä numerona $h(n)$. Voi olla helpompaa keksiä *vertailuehto* $Q(n, n')$ joka kertoo, onko solmu n huonompi, yhtä hyvä tai parempi kuin solmu n'.

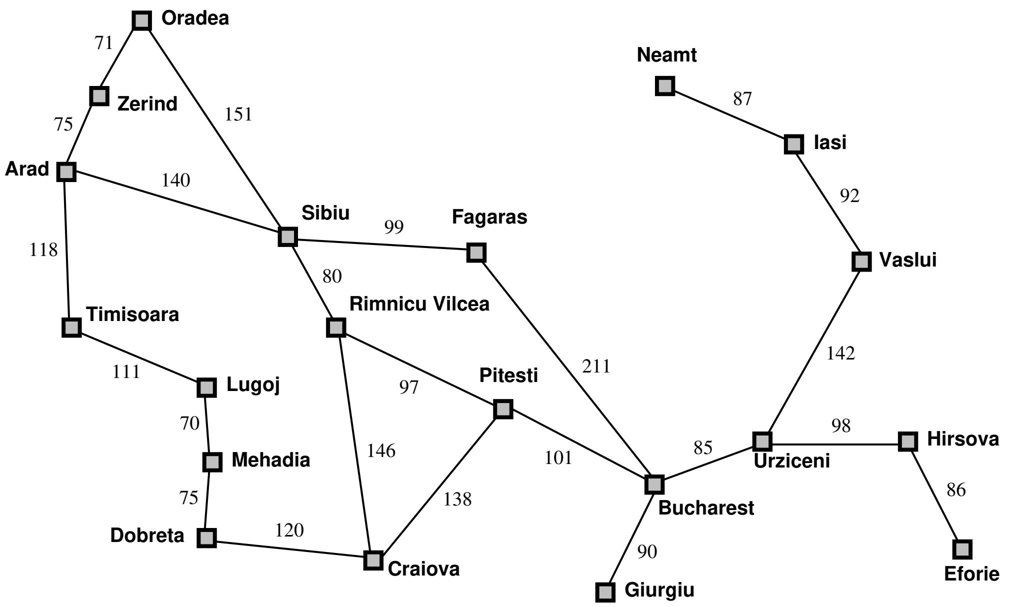
(a) Miten voisimme edelleen tehdä ”paras ensin” -etsintää luvun 3.2 tyyliin?

(b) Entä jos Q voi vastata myös ”en tiedä kumpi on parempi” (joka ei siis ole sama asia kuin ”ne ovat yhtä hyvät”)?

(c) Entä jos haluaisimme minimoida myös polkuhintoja $g(n)$?

(Russell&Norvig, tehtävä 4.4.)

Tehtävä III.4: Simuloi $\mathbf{A}^*$-algoritmia löytämään kilometreissä lyhin maantiereitti Aradista Bukarestiin, kun heuristiikkana on etäisyys kohteeseen linnuntietä:



Straight−line distance to Bucharest	
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

Tehtävä III.5: Unohdetaan hetkeksi se vaatimus, että $\mathbf{A}^*$-algoritmin heuristisen funktion h pitäisi olla luvallinen. Johda sen sijaan sellainen funktio h, että $\mathbf{A}^*$ käyttäytyy kuten

(a) luvun 3.1.1 leveyssuuntainen etsintä

(b) luvun 3.1.2 halvin ensin -etsintä

(c) luvun 3.1.3 syvyyssuuntainen etsintä.

Funktiosi arvo $h(n)$ solmussa n riippukoon vain siihen etsintäpuun juuresta tulevan polun

- hinnasta $g(n)$
- pituudesta (eli puukaarten lukumäärästä) $l(n)$.

Tehtävä III.6: Luennolla sanottiin, että kun etsintäalgoritmin ajo

- päätyi sellaiseen ratkaisuun, jonka syvyys (etäisyys etsintäpuun juuresta) on d
- tuotti puuhun N solmua

niin ajon todellinen haarautumisaste b^* saadaan ratkaisemalla yhtälö

$$N = \frac{(b^*)^{d+1} - 1}{b^* - 1}.$$

Luennolla ei kuitenkaan kerrottu, miten tämä ratkaisu saadaan. Laadi siis algoritmi, joka saa luvut N, d ja $e \in \mathbb{N}$, ja palauttaa vastauksenaan suureen b^* likiarvon e desimaalin tarkkuudella.

(Yhteensä 6 tehtävää.)