Kaj je umetna inteligenca? Je področje ki raziskusje algoritme za raševanje tistega ki rešuje človek.

Kaj je inteligenca? Možnost reševanja problemov.

Človeško obnašanje : Turingov test – človek (vpraševalnik) | računalnik – človek sprašuje in če neve če govori z človekom ali rač. je rač inteligenten? Problem z velikim data setim.

Človeško razmišljanje : kako človek razmišlja in kako to umejuje?

Racionalno razmišljanje : Zahteva znanje nekih dejstev v obliki da lahko sklepamo informacijo z njimi. Verjetno samo z objektivnega pogleda.

Racionalno obnašanje:

Nevarnost glede moralnosti, drugačne vrednosti

Preiskovanje/planiranje

Najdi zaporedij akcij, ki pripeljejo do ciljnega stanja.

Okolje/svet: epizodično(eno za drugom, ne zvezno), samo en agent ki deluje na okolje, popolnjoma vidljivo okolje(vse relavantne informacija za reševanje problema so znane), deterministično(vemo rezultat akcij), statično(množic akcij, ne diskretno?) in znano.

Preiskava algoritma

Neinformirani: Poznamo samo strukture problema

Primer: Iskanje v globino/širino

Informirani: imamo dodatno informacijo – lahko ocenimo razdalje do ciljnega stanja

Primer: A\*

Problemi, planiranje/preiskovanje:

Definicija problema:

* Množica možnih stanj S = {S1,Sˇ2,Sˇ3... }
* Začetek stanja S0
* Množica ciljnih stanj G = {SG1, SG2...}
* Akcijska funkcija : akcija(s) = {a1,a2,a3...} možne akcije za neko stanje ak(S1) = {a1,a3...}

Akcije so oglavnjo neki premiki ki nas premaknejo v drugo stanje? Če je to tranzicijska?? Tranz nam verjetno samo pove v katero stanje bo šlo?

* Tranzicijska akcija: rezultat(a,s) = s'
* Funkcija cene akcije: cena(a,s)->R

Primer: Iskanje najkrajše poti

* Algoritem: iskanje v širini (breadth-first search)

Smo v mestu, akcija nam umogoči premik v drugo mesto ter dostop do drugih akcij (ker smo v drugem kraju in lahko dostopamo drugačne kraje katerih prej nismo mogli)

Predpostavimo da imajo vse akcije enako ceno(1)(velja za vse breadth first search? Verjetno ne, samo breadth first search bo takrat prišlo do optimalne rešitve. Preveč širine in globin je seveda težava. Od globine, št otrokov, kompleksnost narašča exponencialno)

Smo v G, naša fronta(točke o katerih vemo, vozliške katere lahko razvijemo(generirani naslednjiki? Maybe aktualiziramo vse akcije potem ko izberemo vozliške za naslednji korak))

Fronta{Lj, O, T}

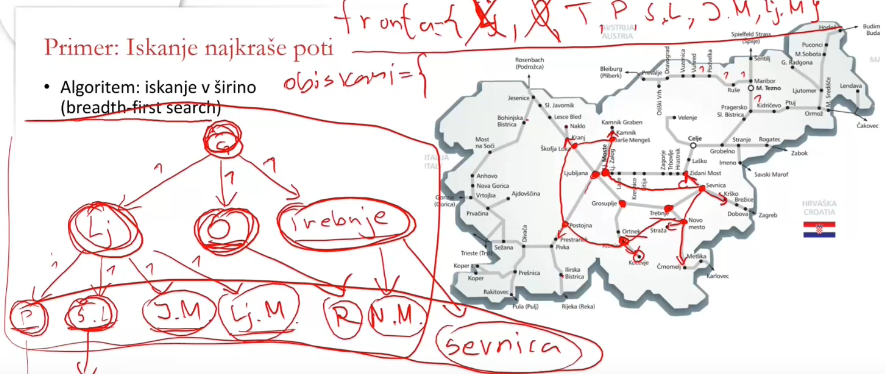
Ko gremo v Lj, lj izbrišemo in dodamo nove lokacije.

Fronta{O,T,P,Š,l,J.M,Lj.M}

Vsak korak ima ceno, ki se veča dalje kot smo od cilja od G ima P ceno 2 saj gre od G proti Lj 1 in iz Lj proti P 2.

Najprej razvijamo najbljižje vozlišča. To pomeni da po G->Lj gremo na O saj ima samo ceno 1, tudi po tem ko gremo v lj. Nato po O gremo v T, po T pa v P itd. Najprej najbližje medtem ko generiramo njihove naslednike.

Note – V množici fronta appendamo naslednjike in popamo obiskane. First in first out npr.



Enak primer

Alogritem: iskanje v globino (depth first search)

Iščemo najglobjega prvo (tisti z največ cene)

Ni zmeram optimalna(seveda) samo da dela.

Je kot breadth samo da ni FIFO , verjetno FILO

Enak primer:

Bolj generalno: Best-First-Search

Akcije lahko imajo različno ceno

Glede na izbiro za f, dobimo več algoritmov.

Hevristika je naš dodatena informacija (pri A\*)