Riešenie problémov opísaných ako hra

problém = hra

- nájsť riešenie problému = nájsť postup, ako (vždy) vyhrať
- situácia sa môže meniť počas hľadania riešenia

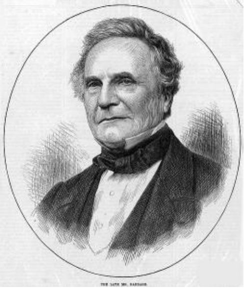
piškvorky

- skúmali oddávna
- Charles Babbage zamýšľal zostrojiť hráča piškvoriek (tic-tac-toe) na svojom analytickom stroji (1834), aby tak získal finančnú podporu pre svoj výskum
- Dvaja hráči hrajú na štvorčekovom hracom poli. Každý hráč má jeden druh symbolu - krúžok alebo krížik. Hráči sa postupne striedajú vo vpisovaní symbolov do políčok. Cieľom je vytvoriť rad piatich* symbolov za sebou - vodorovne, zvisle alebo šikmo. Tento rad musí tvoriť priamku a nesmie byť nikde prerušený súperovým symbolom. Kto takýto rad vytvorí ako prvý, vyhral. Symbol je možné napísať iba na voľné políčka.

  * my budeme v ďalšom uvažovať zjednodušenú verziu s troma symbolmi a hracom poli 3x3.

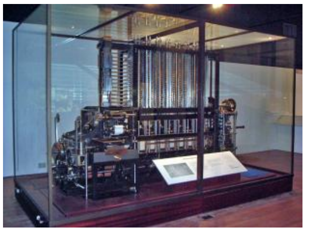
Charles Babbage

- 26 december 1791 Londýn – 18 október 1871 Londýn
- 1810 – Trinity College, Cambridge
- (Klub duchov, Klub vyhadzovačov)
- anglický matematik, filozof, vynálezca, strojný inžinier
- pôvodca myšlienky programovateľného počítača
- rozdielový stroj
- analytický stroj



difference engine

•rozdielový stroj
•výpočet hodnôt polynomiálnych funkcií
•metódou konečných rozdielov
• 1820 - 1. prototyp nedokončil
•neskôr navrhol verziu č. 2
•1989 – 91 Londýnske múzeum vedy dalo postaviť 2 kusy



analytical engine

•analytický stroj
•programovateľný počítač
•dierne štítky (vtedy používané v tkacích stavoch)
• program na diernych štítkoch
•mechanická kalkulačka
•predvídal potrebu sekvencie, selekcie, iterácie
•barónka Ada Lovelace napísala program na výpočet postupnosti Bernoulliho čísel

šach

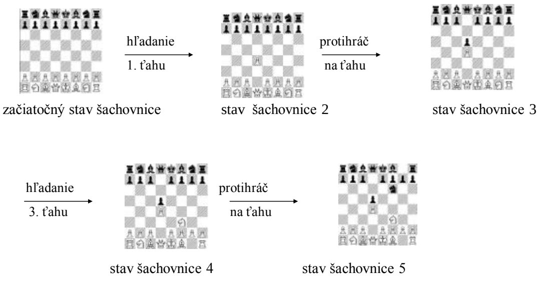
- šach
- Claude Shannon: prvý program, publikoval 1950

Claude Shannon

- April 30, 1916, Petoskey, Michigan – February 24, 2001
- 1936 – Bc elektrotechnika a Bc matematika, U Michigan
- 1937 – MS, MIT, diplomová práca *Symbolická analýza preklápacích obvodov*, označená za najvýznamnejšiu diplomovku storočia
- 1940 – PhD, MIT, Algebra pre teoretickú genetiku (viedol Vannevar Bush)
- 1940 – výskumník Princeton
- americký matematik, elektroinžinier, kryptograf
- cez 2. svetovú vojnu - Bellove laboratóriá
- 1948 - matematická teória komunikácie
- teória informácie, entropia

7

šach – hľadanie najlepších ťahov



8

šach

- Claude Shannon: prvý program, publikoval 1950
- 1957 – Newell a Simon predpovedali, že počítač sa stane majstrom sveta do 10 rokov
- 1958 – IBM 704 prvý počítač, hrajúci šach
- 1967 – Mac Hack dokázal sa zúčastniť turnaja
- 1983 – Belle získalo status šachového majstra (2250 bodov), Bell Labs, počítač PDP 11
- ½ 80' – CMU začal vývoj toho (Chip Test, Hlboká myšlienka), čo sa neskôr preslávilo ako Hlboká modrá.
- 1989 – projekt prešiel k IBM (veľká modrá firma)
- 1997.5.11 – Garry Kasparov prehral 2.5:3.5

9

dáma

- ~5 x 10^{20} možných pozícií
- 1952 Arthur Samuel program pre IBM 701
- 1954 Arthur Samuel program pre IBM 704
  - (10k hlavnej pamäti)
  - mechanizmus učenia sa (učí sa vlastnú vyhodnocovaciu funkciu)

- dlho nič – poklona Samuelovi

10

dáma

- 1989 - Jonathon Schaeffer: Chinook
  - databáza otvorení
  - hľadanie v strednej hre
  - databáza koncoviek
- 1992 – Chinook vyhral turnaj
- 1994 – aj proti majstrovi sveta
- 2007 - Jonathon Schaeffer et al. v Science:
  - Dáma je vyriešená!
  - Perfektná hra oboch hráčov zaručuje remízu.

11

hra

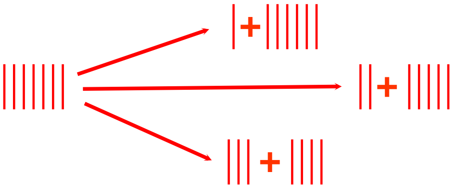
- hranie hry:
- súper sa neustále snaží zmariť každý ťah súpera

- 1944 – John (János) von Neumann navrhol metódu hľadania riešenia:
  - maximalizuje pozíciu hráča a minimalizuje pozíciu protihráča (odtiaľ minimax)

12

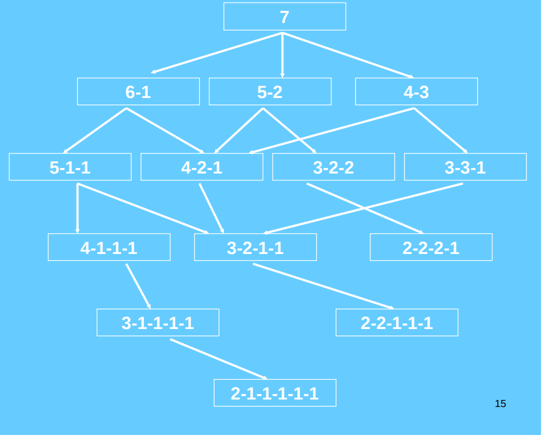
príklad nim

- jednoduchá hra (pochádza zo starovekej Číny, fan-tan, zápalky)
- začína s jednou kôpkou paličiek
- v každom kroku hráč musí vybrať kôpku tak, že rozdelí paličky do dvoch neprázdnych nerovnakých kôpok



príklad nim

- ak hráme hru len so 7 zápalkami/paličkami, celý priestor je dosť malý na to, že môžeme nakresiť úplný strom hry
- kvôli úspore nakreslíme graf hry (jeden stav sa reprezentuje len raz)



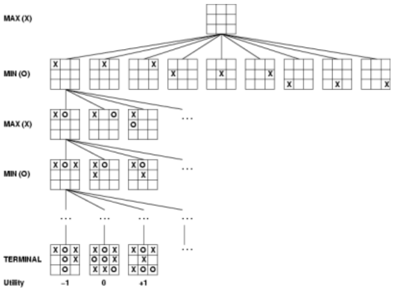
graf hry – čo s ním?

- Čo s grafom hry? Ako nám pomôže pri rozhodovaní, ako hrať (a vyhrať)?
- poďme na minimax
- potrebujeme ohodnotenie pozície:
  - funkcia užitočnosti, vyhodnocovacia funkcia, atď.
- pomenujme hráčov Min a Max

vyhodnocovacia funkcia f()

- vyhodnocovacia (bodovacia) funkcia f() dáva vyššie hodnoty pre lepšie pozície z pohľadu Maxa
- predpokladajme:
  - 1 – výhra pre Maxa
  - 0 – výhra pre Mina
- ide iba o porovnávanie hodnôt
- pri inej hre môže byť
  - 1 – výhra
  - 0 – remíza
  - -1 – prehra

zjednodušené piškvorky: strom hry



minimax – základná myšlienka

- hráč Max si vyberá najlepší dostupný ťah
  - čiže vyberie ťah vedúci do nasledujúceho stavu s najvyššou užitočnosťou
  - čiže hodnota uzla Maxa je maximum spomedzi hodnôt nasledujúcich možných stavov
  - čiže maximum nasledovníkov uzla v strome hľadania

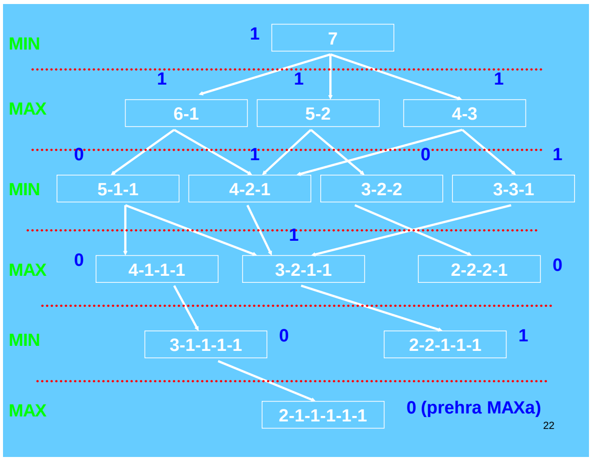
minimax – základná myšlienka

- hráč Min si tiež vyberá najlepší dostupný ťah
  - čiže najhorší dostupný z pohľadu Maxa
  - čiže vyberie ťah vedúci do nasledujúceho stavu s najnižšou užitočnosťou
  - čiže hodnota uzla Mina je minimum spomedzi hodnôt nasledujúcich možných stavov
  - čiže minimum nasledovníkov uzla v strome hľadania

minimax – základná myšlienka

- priebeh hry:
- hráči Max a Min sa striedajú,
  - začína Min
- je to jedno, kto začína, podstata zostáva

- Treba vygenerovať úplný Min/Max strom
- predpokladajme:
  - 1 – výhra pre Maxa
  - 0 – výhra pre Mina



zhodnotenie vyhodnotenia stromu hry

- začiatočný uzol Min má hodnotu +1
- všetky ťahy Mina vedú do stavov s hodnotou +1 pre Maxa
- Min nedokáže zabrániť prehre
- z ohodnotení uzlov v strome sa dá odčítať najlepší ťah v každom kroku

algoritmus minimax

```
function MiniMax-Rozhodnutie(stav) returns operátor
  v ← Max-Hodnota(stav)
     return op v Nasledovníky(stav) s hodnotou v
function Max-Hodnota(stav) returns bodová-hodnota
  if cieľový-Test(stav) then  return Bodovacia-Funkcia(stav)
  v ← -∞
  for each s in Nasledovníky(stav) do
     v ← Max(v, Min-Hodnota(s))
  end
function Min-Hodnota(stav) returns bodová-hodnota
  if cieľový-Test(stav) then return Bodovacia-Funkcia(stav)
  v ← +∞
  for each s in Nasledovníky(stav) do
     v ← Min(v, Max-Hodnota(s))
  end
return v
```

vlastnosti algoritmu minimax

- Úplný? Áno (ak je strom konečný)
- Optimálny? Áno (proti optimálnemu protihráčovi)
- Časová zložitosť? $O(b^h)$
- Pamäťová zložitosť? O(bh) (hľadá sa do hĺbky)
- Šach: b ≈ 35, h ≈100 pri "rozumných" hrách
  → presné riešenie je úplne neschodné

zhodnotenie minimaxu

- pekné, ale...
- skutočné hry majú stromy hľadania omnoho širšie a hlbšie než nim
- nie je možné ohodnotiť celý strom
- čo s tým?
  - stanoviť hranicu, do akej hĺbky sa hľadá
  - koncový stav nie je koncový (kde skončí hra výhrou/prehrou), je koncový (kde došlo k useknutiu)

ohraničený minimax

- koncové stavy už nebudú istá výhra/prehra
  - v skutočnosti budú, len to nevieme s dostupnými výpočtovými zdrojmi rozhodnúť
- musí sa heuristicky/približne ohodnotiť kvalitu koncových stavov
- vyhodnotenie môže byť náročné (na vymyslenie a potom aj na čas výpočtu), najmä pre prípady nie jasnej výhry/prehry
- nasleduje umelý príklad ohraničeného minimaxu

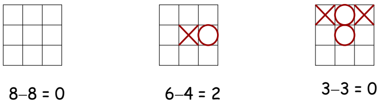
heuristická vyhodnocovacia funkcia

- Funkcia e: stav s → číslo e(s)
- e(s) je heuristika, ktorá odhaduje, ako nádejný je stav s pre Maxa
  - e(s) > 0 znamená, že stav s je nádejný pre Maxa (čím vyššia hodnota, tým nádejnejší/lepší)
  - e(s) < 0 znamená, že stav s je nádejný pre Mina
  - e(s) = 0 znamená, že stav s je neutrálny

príklad e_{nim}

$e_{nim}(s)$ =

počet riadkov, stĺpcov a uhlopriečok voľných pre Maxa
- (mínus)
počet riadkov, stĺpcov a uhlopriečok voľných pre Mina



8–8 = 0 6–4 = 2 3–3 = 0

ako navrhnúť vyhodnocovaciu funkciu?

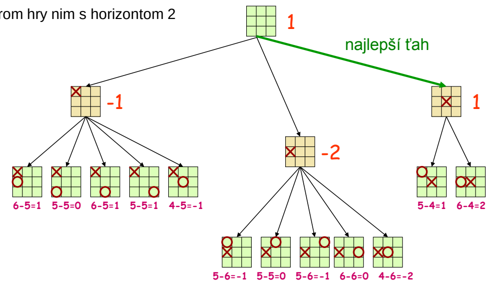
- zvyčajne ako vážený súčet "čŕt":

$$e(s)=\sum_{i=1}^{n} w_i f_i(s)$$

- Črty f_i môžu zahŕňať
  - počet kameňov každého možného druhu
  - počet možných ťahov
  - počet ovládaných políčok

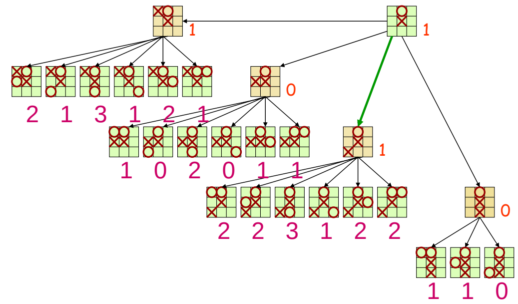
príklad e_{nim}

strom hry nim s horizontom 2



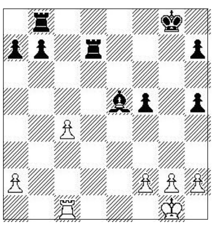
31

príklad e_{nim} pokračovanie



32

príklad šachovej heuristickej funkcie



- čierny má:
  - 5 sedliakov, 1 strelca, 2 veže
- skóre = 1*(5)+3*(1)+5*(2)
  = 5+3+10 = 18

biely má:
  - 5 sedliakov, 1 vežu
- skóre = 1*(5)+5*(1)
  = 5 + 5 = 10

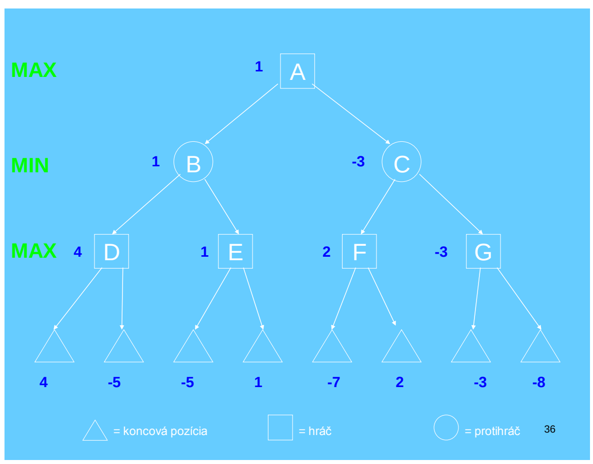
celkové ohodnotenie pozície:
čierny = 18-10 = 8
biely = 10-18 = -8
e(pozícia) = -8

33

Zapamätávanie hodnôt zdola nahor. Prečo?

- pri každom vnútornom uzle N sa zapamätá hodnota najlepšieho stavu, ktorý môže Max dosiahnuť v hĺbke h, ak Min hrá svoju najlepšiu hru (podľa rovnakého kritéria ako Max)
- takáto zapamätaná hodnota je lepším odhadom nádejnosti stav(N) než e(stav(N))

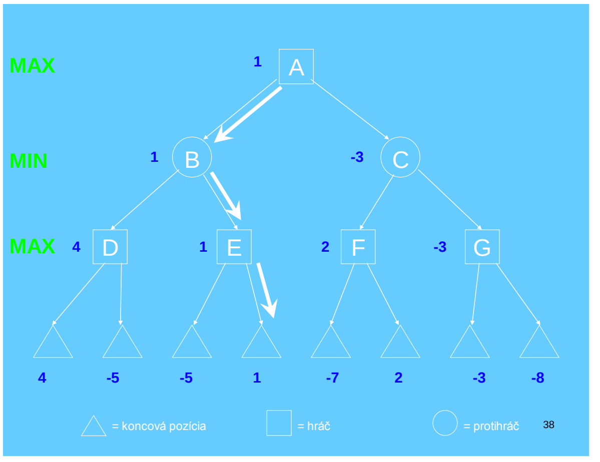
34



príklad: ohraničený minimax

- ak hrajú oba hráči svoje najlepšie ťahy, ako bude prebiehať hra?

37



príklad: ohraničený minimax

- dokonalá hra vedie do koncového uzla s rovnakým ohodnotením, ako má začiatočný uzol
- aj všetky uzly po ceste hry majú také isté ohodnotenie
- v podstate, to je význam hodnoty koreňa

39

ohraničený minimax

- minimax do stanovenej hĺbky:
- treba vytvoriť strom hry a potom popridávať minimaxové hodnoty, aby sme zistili, ako sú ohodnotené ťahy zo súčasnej pozície.

40

prečo minimax nestačí?

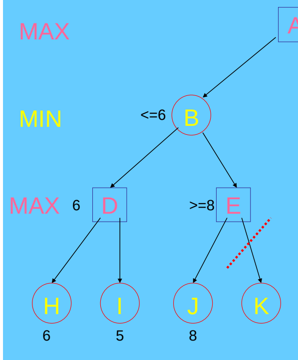
- efektívnosť hľadania
  - stromy hry sú veľmi veľké
  - vyhodnocovanie pozícií je náročné a dlho trvá
- ako sa dá zmenšiť počet vyhodnocovaných uzlov?
- ohraničovanie hľadania do hĺbky má slabiny
  - prečo?
- ako ohraničovať rozrastanie prehľadávanej časti stromu do šírky?
  - α/β hľadanie

41

prečo minimax nestačí?

- hodnoty sme pridávali zdola nahor od listov po prehľadaní do šírky
- nevýhoda: vyžaduje mnoho pamäti
  - spomeňme si na rozdiel v pamäťovej náročnosti hľadania do šírky a do hĺbky
- minimax pri prehľadávaní stromu hry do hĺbky by potreboval omnoho menej pamäti
- minimax sa naozaj dá urobiť aj pri prehľadávaní do hĺbky
- ale: zníženie potreby pamäti nie je jediná vyhoda:

42



STOP! Čo sa dá ešte usudzovať!?

keď odhalíme, že e(D) = 6

usúdime, že e(B) <= 6

keď odhalíme, že e(J) = 8

usúdime, že e(E) >= 8

rozvíjanie E sa môže skončiť, pretože najlepšia hra nepôjde cez E

hodnota K je irrelevantná – odsekni ho!

= hráč = protihráč

ako zlepšiť minimax?

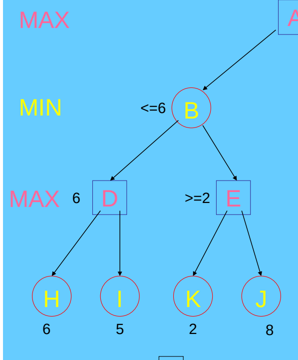
- ak prehliadame strom do hĺbky, nemá zmysel vyhodnocovať uzol K.
- nech by bola hodnota K hocijaká, žiadna hra cezeň nepôjde
- uzol K možno odseknúť, t.j. nebude sa rozvíjať
- v (stave reprezentovanom) uzle B, lebo Min nikdy nezvolí E; pretože J je lepší než D pre Maxa, Min mu nesmie umožniť takú príležitosť
- zdá sa, že ide o úsporu len jedného uzla. Ale K môže nachádzať ešte vysoko nad stanovenou hranicou hĺbky hľadania. Vtedy sa usekne celý podstrom s počtom uzlov exponenciálne závislým od jeho hĺbky

zlepšenie minimaxu

- Predpokladajme, že by sme robili hľadanie do šírky. Mohli by sme takto osekávať?
- Nie! Osekávanie sa opiera o to, že sme už vyhodnotili uzol D na základe vyhodnotenia podstromu od ním.
- Takýto spôsob osekávania je príkladom alfa-beta osekávania. Opiera sa o hľadanie do hĺbky so stanovenou obmedzenou hĺbkou

zlepšenie minimaxu

- predpokladajme, že
  - uzly K a J by sa vyhodnocovali v opačnom poradí
  - dalo by sa podobne usekávať?
- závisí od hodnoty K
- predpokladajme, že K dostane hodnotu 2 a rozvíja sa skôr:



STOP! Čo sa dá ešte usudzovať!?

keď odhalíme, že e(D) = 6

usúdime, že e(B) <= 6

keď odhalíme, že e(K) = 2

usúdime, že e(E) >= 2

rozvíjanie E sa nemôže skončiť, pretože najlepšia hra stále ešte môže ísť cez E

Hodnota J je relevantná – nesekáme

= hráč = protihráč

zlepšenie minimaxu

- Ak K dostalo hodnotu 2 a rozvíjalo sa skôr (ako J), nenastali podmienky na useknutie nasledovníka E.
- Aby sa dalo čo najviac usekávať, treba začať rozvíjať tie nasledovníky, ktoré sú najlepšie pre ich predchodcu
  - to však nevieme
  - zišla by sa heuristika usporadúvajúca nasledovníky
- ak by sme to vedeli, hľadanie by mohlo ísť do väčšej (až dvojnásobnej) hĺbky
  - napr pri šachu by to znížilo faktor vetvenia z 30 na 8
  - čo je významné zlepšenie!

zlepšenie minimaxu

- Uvažované hry sú symetrické. Čo sme uvažovali usekávať z pohľadu Maxa, môžeme podobne aj z pohľadu Mina.
- zdôvodnenie rovnaké, len roly Maxa a Mina sa vymenia.
- Odhalíme ďalšie uzly, ktoré nemôžu byť súčasťou najlepšej hry (a možno ich useknúť)

49