Artificial Intelligence:

- Intelligence
  - ability to understand
  - problem solving
  - trial & error learning (experience based)
  - + Intuition, awareness ?
  - + Creativity
  - decision making
  - search
  - ability to reason (deductive, inductive)
  - Analogy

AGENT = Program + Archi.

(aeger)

to do

Five Tribes (AI Algorithms)

- Evolutionaries (Evolving Structure)
- Connectionists (Learning Parameters)
- Symbolists (composition of elements)
- Bayesians (Weighing evidences)
- Analogizers (Mapping to new situation)

– Master Algorim

Thinking Humanly

- machines with mind

+ Cognitive Science

Acting Humanly

+ Turing Test

- make comp do things at which, at the moment, people are better

Thinking Rationally

- study of mental faculties through the use of comp models

+ Laws of thought, Game Theory

Acting Rationally

- study of design of intelligent agents

+ Correct Inference

AI Foundations:

- Philosophy
- Mathematics
  - Gödel Incompleteness
  - D Hofstader (GEB)
  - Uncertainty
  - Turing computa.
  - Tractability
- Economics - max payoff - far in future
- Neuroscience
- Psychology
- CE + buid eff comp
- Control - Cybernatics
- Linguistics

Vision

Speech

Smell

⋮

sensor

- Knowleje Represention
- Memory
- IR
- CV
- SP/R

- Search
- Recognition
- Reasoning
- Planning
- Problemsolving

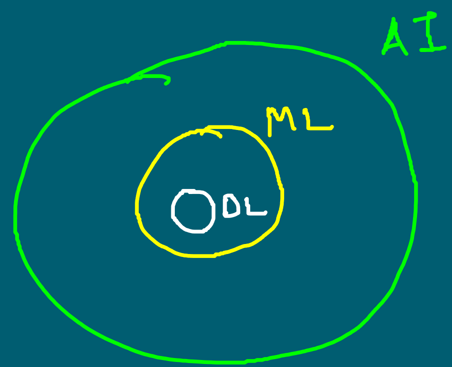
ML

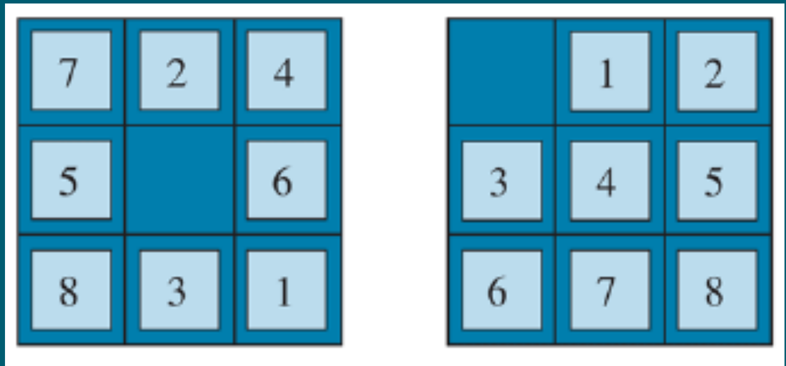
Robotic Control

Graphics

Speech synthesis

Sense ⟶ Deliberate ⟶ Act

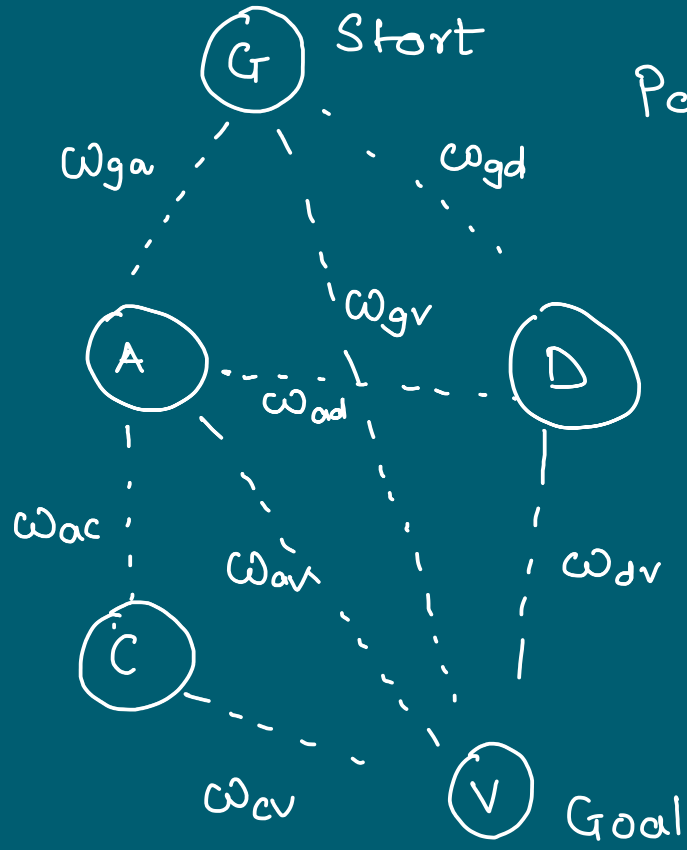




Start State   Goal State

8 - Puzzle

Path Search



Missionaries & Cannibals

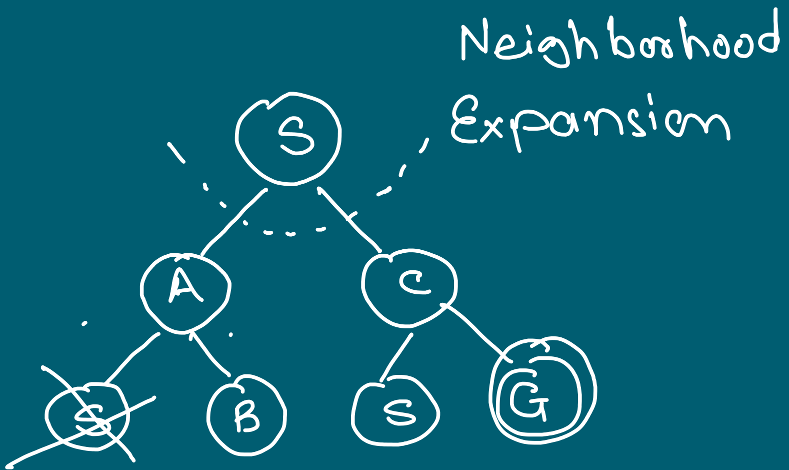
$M_1 M_2 M_3$

$C_1 C_2 C_3$

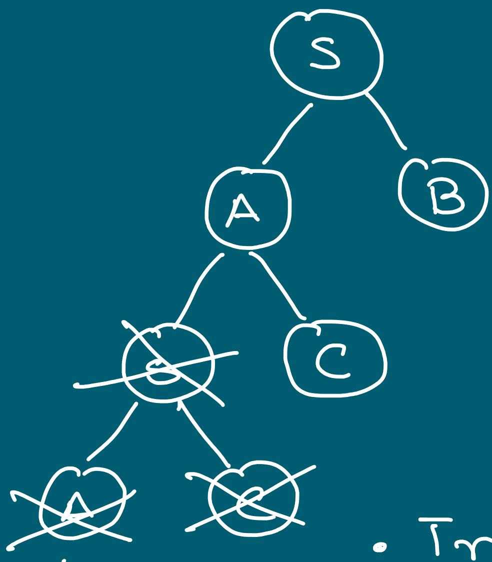
- Initial Configuration ⟶ End Configuration
- Tree - Graph traversals

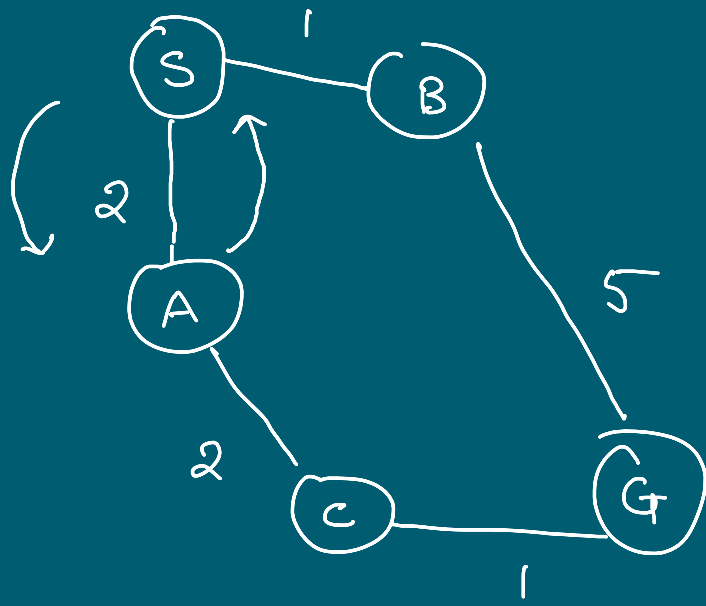
- Breadth / Depth First Search

BFS



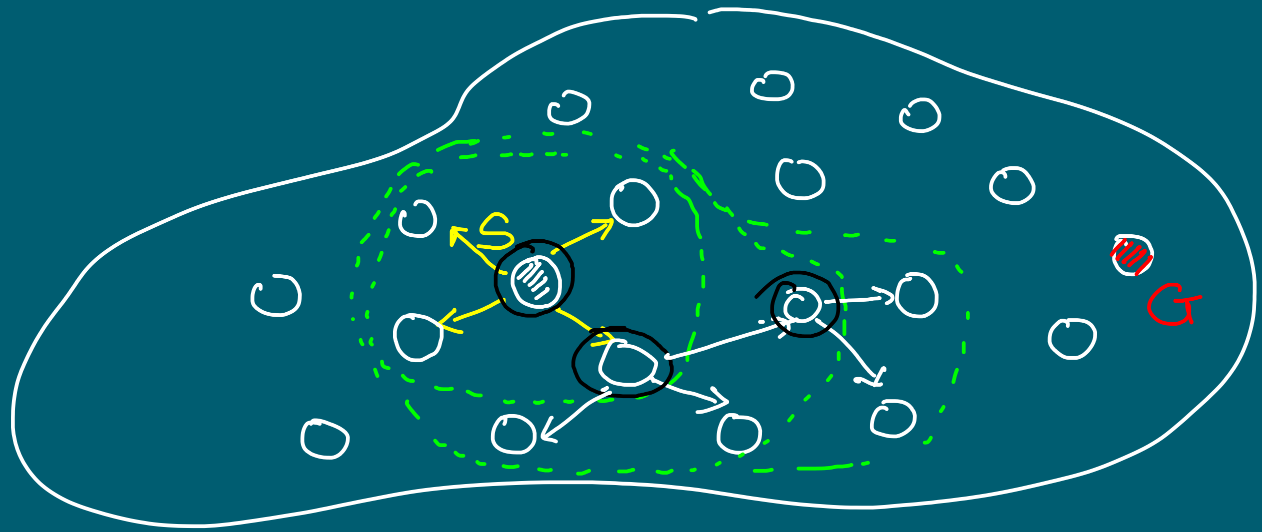
DFS





- start node
- State Transition
- Adjacency matrix/list
- List to keep track of visited nodes / Hashtable
- Set of nodes to be explored in a queue [FIFO | STACK]
- Goal Test

State Space Search:



— Frontier

— Visited

State vs Node

node.state

node.parent

node.action

node.path_cost

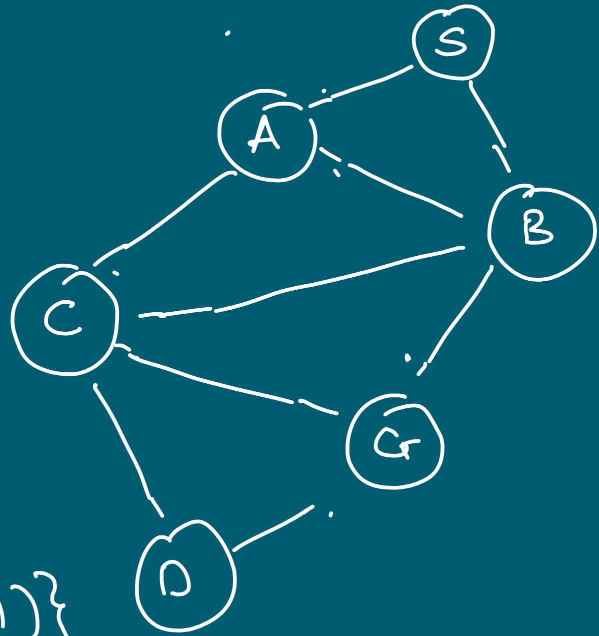
Frontier

pop, top

is_empty

add

Best First Search



Init

~~(S, ϕ, 0)~~

~~(A, C, 3)~~

~~(B, C, 3)~~

~~(G, C, 3)~~

(D, C, 3)

Frontier

{ ~~(S, ϕ, 0)~~ }

{ ~~(A, S, 1)~~, (B, S, 1) }

{ ~~(B, S, 1)~~, (C, A, 2) }

(G, B, 2) }

{ $\boxed{\text{(G, B, 2)}}$

(D, C, 3) }

(Explored!)

Visited

{ }

{ (S, ϕ, 0) }

{ (S, ϕ, 0), (A, S, 1) }

{ (S, ϕ, 0), (A, S, 1), (B, S, 1) }

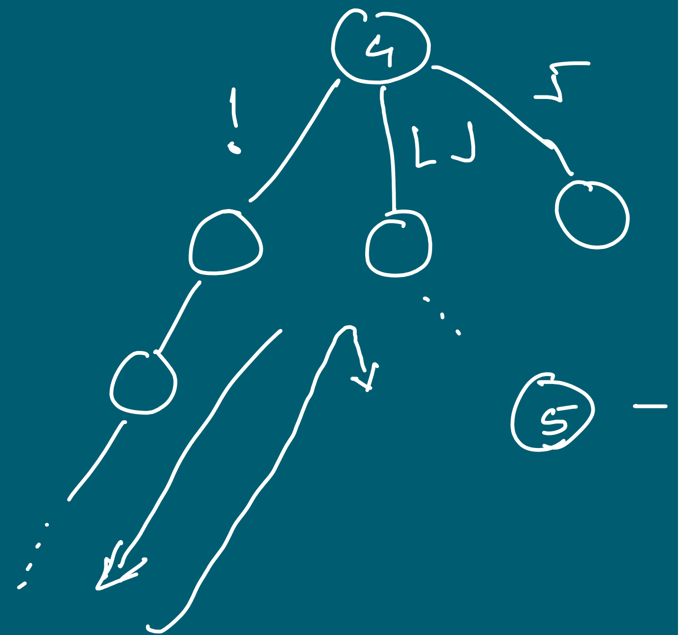
{ (S, ϕ, 0), (A, S, 1) (B, S, 1)

(C, A, 2) }

Search Performance

1. Completeness - Is the algorithm guuranteed to find 'a' solution when there is one, and to correctly report failure when there is not?

BFS DFS Best FS

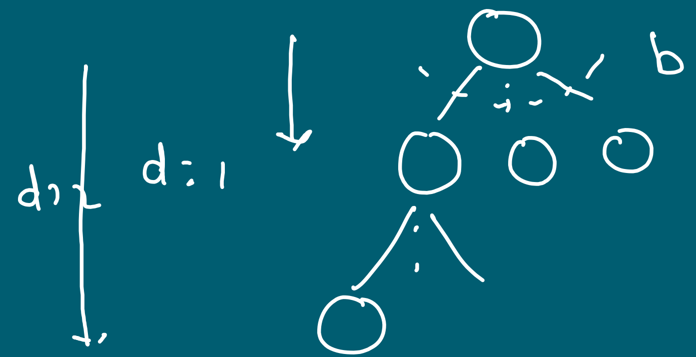


2. Optimality - lowest path cost solution - - -

3. Time complexity 4. Space Complexity

1 Million nodes / sec , 1000 bytes / node

Branching factor : $b = 10$



depth	Nodes	Time	Mem
2	110	.11 ms	107 KB
4	11,110	11 ms	10.6 MB
6	10^6	1.1 s	1 GB
8	10^8	2 min	103 GB
10	10^{10}	3 hrs	10 TB
12	10^{12}	13 days	1 PB
14	10^{14}	3.5 years	99 PB
16	10^{16}	350 years	10 EB

Missionaries & Cannibals

$M_1 M_2 M_3$

$C_1 C_2 C_3$

✓

L | R

CM CC $\xrightarrow{?}$

L

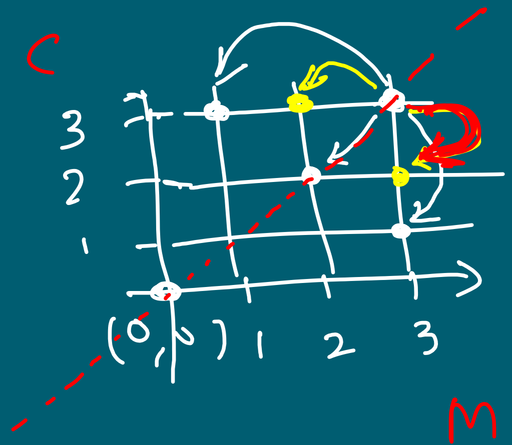
$M_1 M_2 M_3$ — MMM

↓

3M

S $\quad$ G

$(3,3) \rightsquigarrow (0,0)$



Frontier $\qquad$ Visited

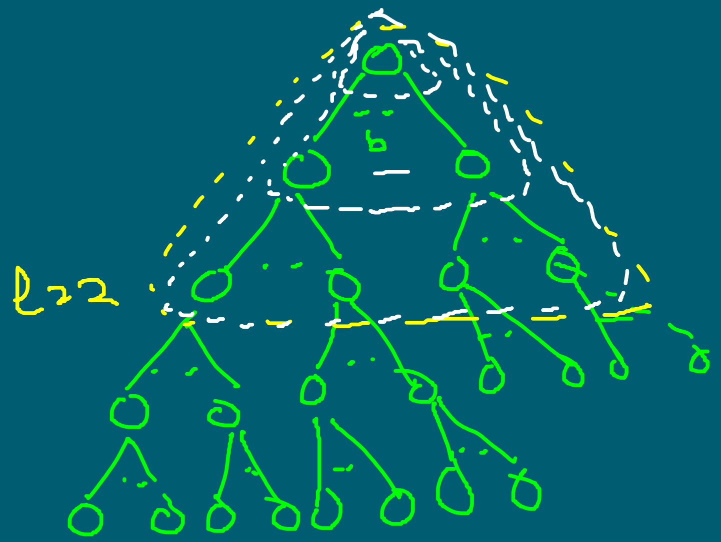
$((3,3),\phi,0)$ $\quad$ $\{(3,3),\phi,0\}$ $\quad$ $\{\ \}$

$\{((3,2),(3,3),1)\}$ $\quad$ $\{((3,3),\phi,0)\}$

↓

Uninformed Search

- Use of info obtained from env
- Blind search – BFS / DFS
- Best first (to root) $\longrightarrow$ Uniform Cost Search
- Depth Limited Search
- Iterative Deepening Search



l	0	1	2	3	...	d
#	1	$1+b$	$1+b+b^2$	$1+b+b^3$		$\underline{1+b+\dots+b^d}$

$$\# \mathrm{IDS}(d) = d + (d-1)b + (d-2)b^2 + \dots + b^d \sim O(b^d)$$

$$\# \mathrm{BFS}(d) = 1 + b + \dots + b^d \sim O(b^d)$$

$b = 10, \; d = 5$

$\# \mathrm{IDS} = 1,23,450$

$\# \mathrm{BFS} = 1,11,110$

Criterion	Breadth-First	Best First Uniform-Cost	Depth-First	Depth-Limited	Iterative Deepening	Bidirectional (if applicable)
Complete?	Yes[1]	Yes[1,2]	No	No	Yes[1]	Yes[1,4]
Optimal cost?	Yes[3]	Yes	No	No	Yes[3]	Yes[3,4]
Time	$O(b^d)$	$O(b^{1+\lfloor C^*/\epsilon \rfloor})$	$O(b^m)$	$O(b^\ell)$	$O(b^d)$	$O(b^{d/2})$
Space	$O(b^d)$	$O(b^{1+\lfloor C^*/\epsilon \rfloor})$	$O(bm)$	$O(b\ell)$	$O(bd)$	$O(b^{d/2})$

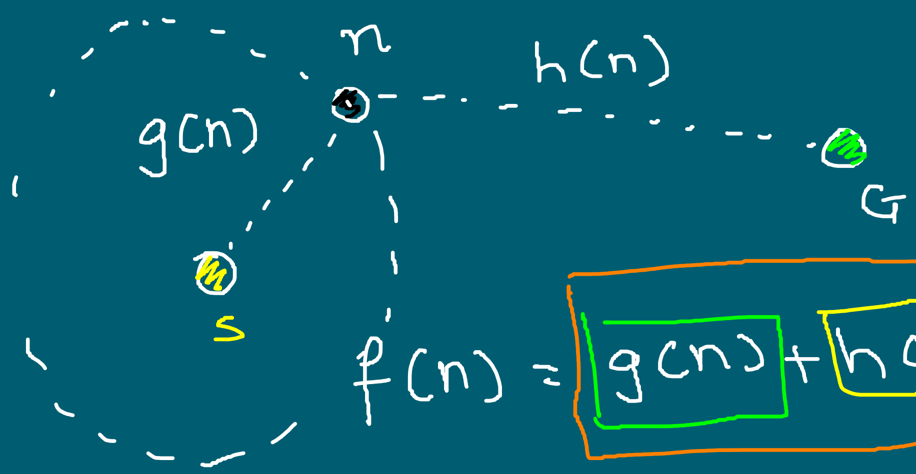
Evaluation of search algorithms. b is the branching factor; m is the maximum depth of the search tree; d is the depth of the shallowest solution, or is m when there is no solution; ℓ is the depth limit. Superscript caveats are as follows: [1] complete if b is finite, and the state space either has a solution or is finite. [2] complete if all action costs are $\geq \epsilon > 0$; [3] cost-optimal if action costs are all identical; [4] if both directions are breadth-first or uniform-cost.

RN

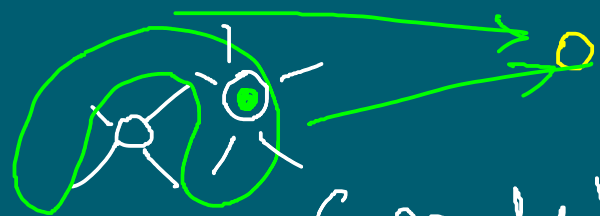
Informed Search

- $\boxed{\text{Best First Search}}$
- $\boxed{\text{Hill Climbing}}$
- $\boxed{A^* \quad \text{(A-star)}}$

distance to goal

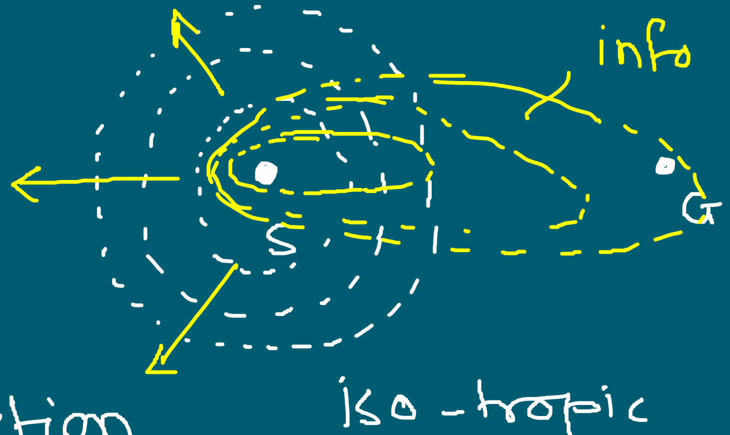


$$f(n) = \boxed{\boxed{g(n)} + \boxed{h(n)}}$$



Greedy!

Search Infrastructure

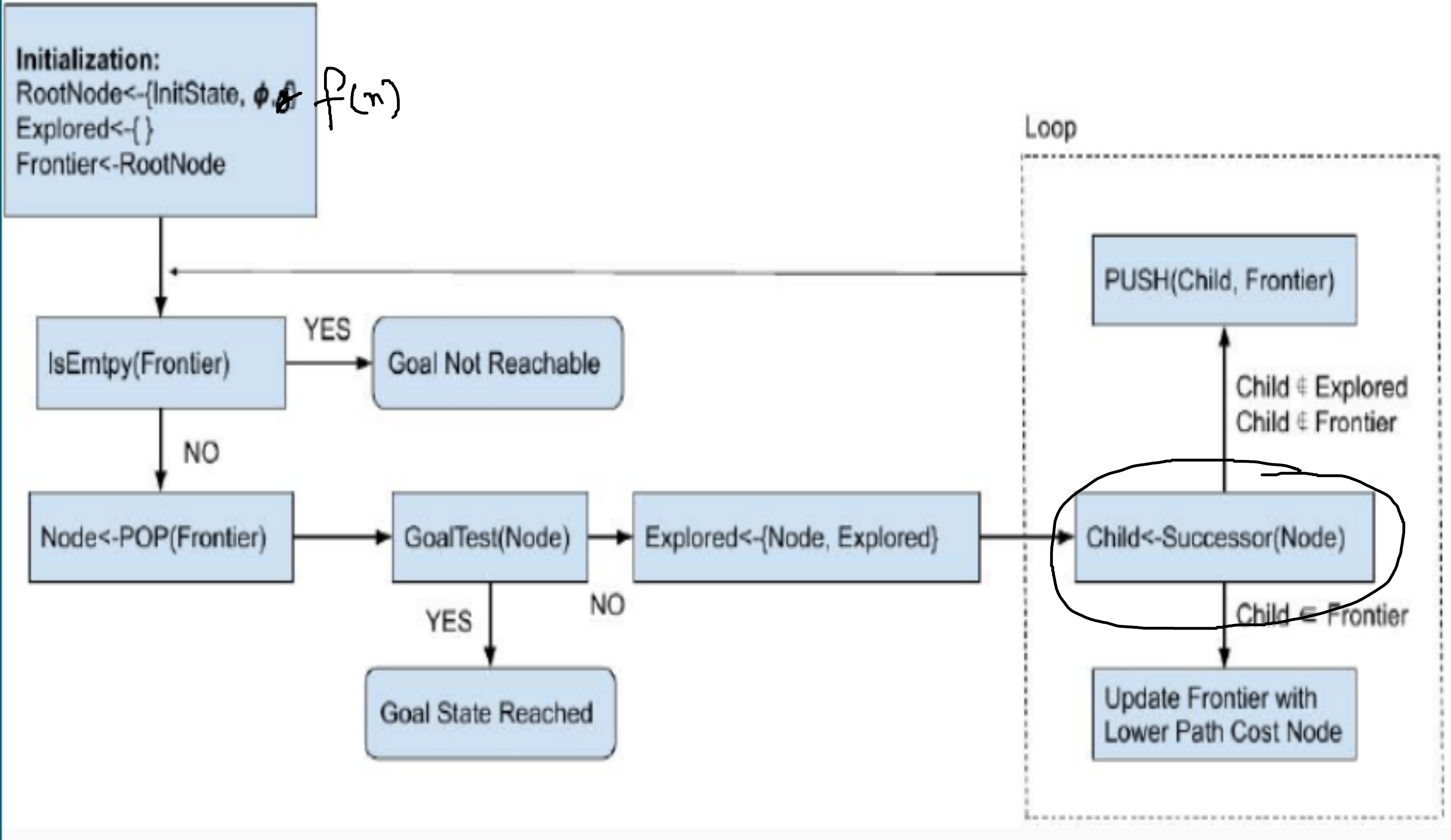


- Problem: Initial State, GoalTest,
  Transition(successor)

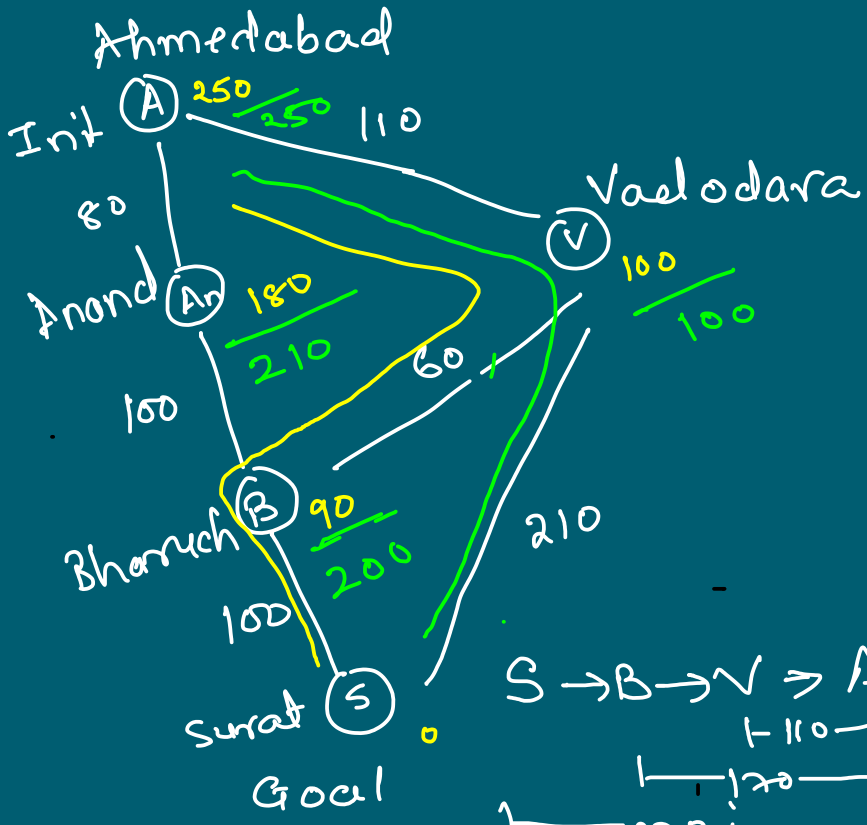
- Agent: Node - state, parent, action,
  path cost / evaluationfun → f(n)

  Explored set (visited) - List / Hash table

  Frontier : Queue - (FIFO, LIFO, Priority)



Initialization:
RootNode<-{InitState, ϕ,ϕ
f(n)
Explored<-{ }
Frontier<-RootNode
Loop
PUSH(Child, Frontier)
IsEmtpy(Frontier)
YES
Goal Not Reachable
Child ∉ Explored
Child ∉ Frontier
NO
Node<-POP(Frontier)
GoalTest(Node)
Explored<-{Node, Explored}
Child<-Successor(Node)
YES
NO
Child ∈ Frontier
Goal State Reached
Update Frontier with
Lower Path Cost Node



$S \to B \to V \to A$

$\vdash 110 \dashv$

$\vdash 170 \dashv$

$\vdash 270 \dashv$

$$f(n) = g(n) + \omega \cdot h(n)$$

f

$\omega = 0$	Dijkstra / Unif cost
	Best FS
$\omega = 1$	A^*
$\omega = \infty$	Greedy Best FS
$1 < \omega < \infty$	weighted A^*

$\omega = 0$

Frontier	$\{(A, \phi, 0)\}$	$\{(An, A, 80), (V, A, 110)\}$	$\{(V, A, 110), \boxed{(B, An, \cancel{180})}\}$	$\{(B, V, 170), (S, V, 320)\}$	$\boxed{\{(S, B, 270)\}}$
Visited	$\{\}$	$\{(A, \phi, 0)\}$	$\{(A, \phi, 0), (An, A, 80)\}$	$\{(A, \phi, 0), (V, A, 110), (An, A, 80),\}$	$\{(A, \phi, 0), (V, A, 110), (An, A, 80), (B, V, 170)\}$

$\omega = 1$

Frontier	$\{(A, \phi, 250)\}$	$\{(An, A, 260) \; 290, (V, A, 210) \; 260\}$	$\{(An, A, 260), (B, V, 260), \boxed{(S, V, 320)} \; 380\}$	$\{(B, V, 260), (S, V, 320)\}$	$\boxed{\{(S, B, 270)\}}$
Visited	$\{\}$	$\{(A, \phi, 250)\}$	$\{(A, \phi, 250), (V, A, 210)\}$	$\{(A, \phi, 250), (V, A, 210)\}$	$(An, A, 260), (B, V, 260)\}$

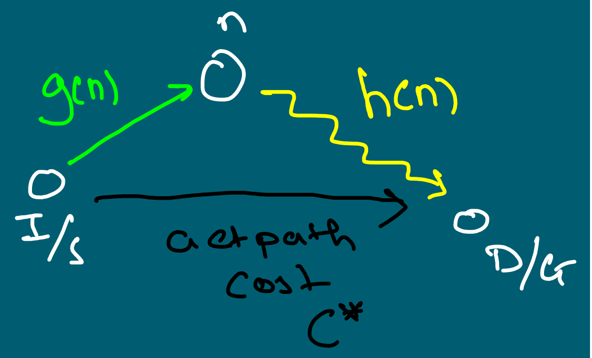
1964 Nilson improves Dijkstra's algo, "invented" heuristic based approach — A1

1967 B Raphael — A2

1968 Peter E Hart — A^* (Duda & Hart Pattern Classification)

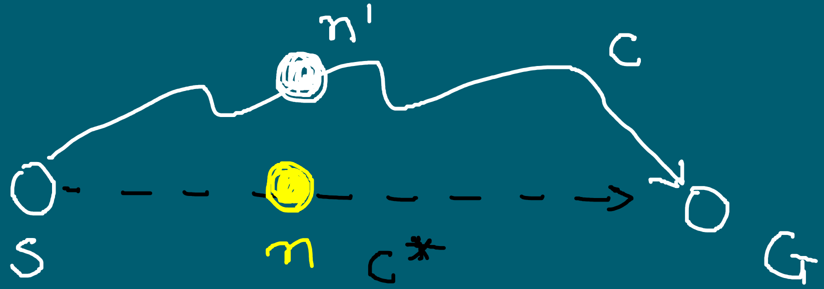
Heuristic Function

- Admissible - it never overestimates the cost to reach a goal.

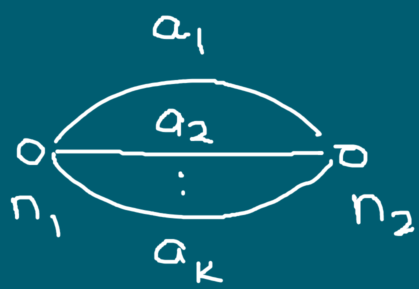


- h - admissible $\implies$ A^* is optimal

$$f(n) = g(n) + h(n) < C^*$$



$c(n_1, a_i, n_2)$



- A^* returns a suboptimal path given that h is admissibl

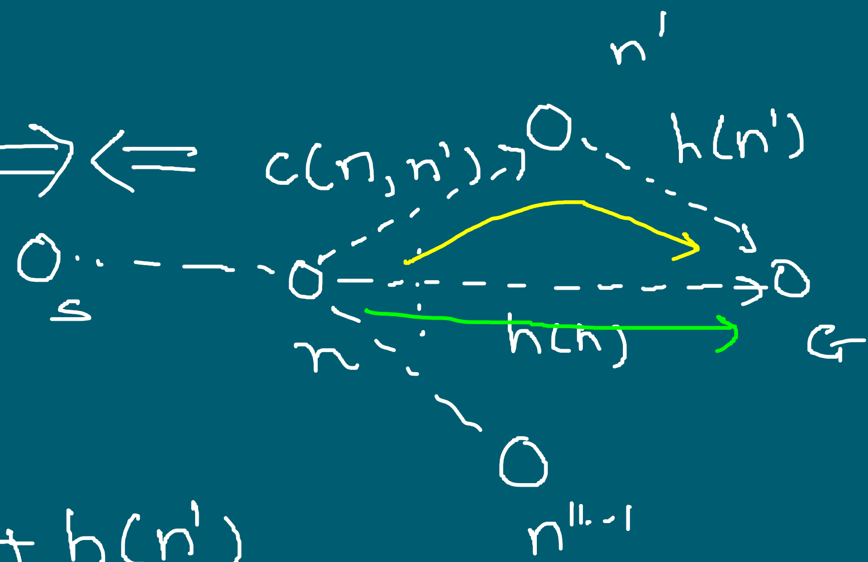
$$f(n) = g(n) + h(n)$$

$$f(n) \geq f(n')$$

$$f(n) \leq c^*$$

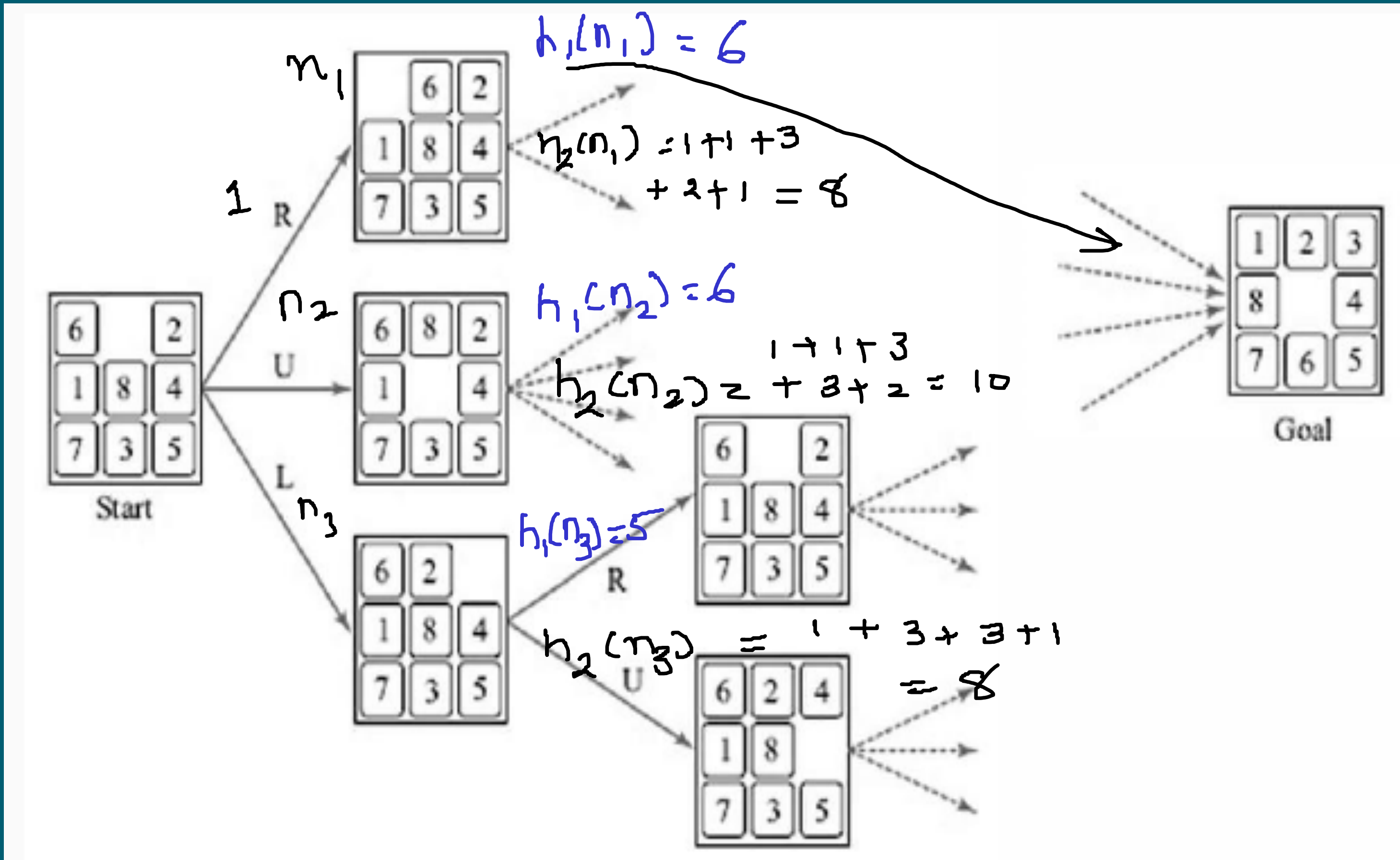
$$g(n) + h(n) \geq C > c^*$$

$$h(n) > c^* - g(n) \quad \Rightarrow\Leftarrow$$



- Slightly Stronger: Consistency

$$h(n) \leq c(n, n') + h(n')$$



https://www.sfu.ca/~jtmulhol/math302/notes/permutation-puzzles-book.pdf

S [6 0 2 1 8 4 7 3 5]

G [1 2 3 8 0 4 7 6 5]

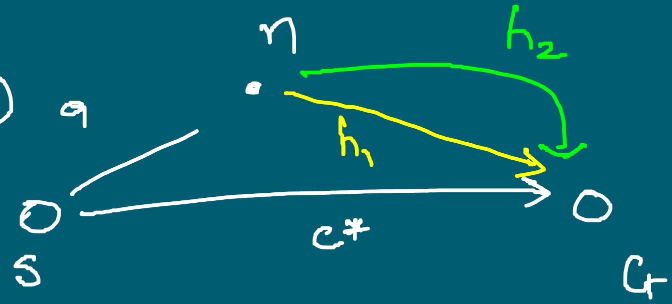
- 9!

- Alternating Group
- Transposition

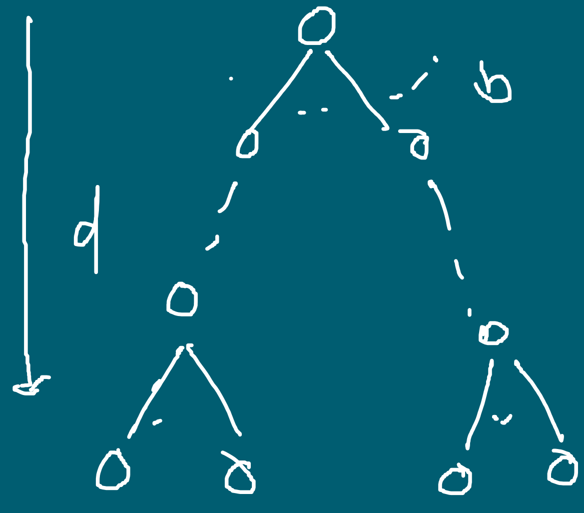
$h_1(n)$ = # misplaced tiles, $h_2(n) = \sum$ Manhattan Distance

$$h_1(n) \le h_2(n) < C^* - g(n),$$

both (admissible)



	Search Cost (nodes generated)			Effective Branching Factor		
d	BFS	$A^*(h_1)$	$A^*(h_2)$	BFS	$A^*(h_1)$	$A^*(h_2)$
6	128	24	19	2.01	1.42	1.34
8	368	48	31	1.91	1.40	1.30
10	1033	116	48	1.85	1.43	1.27
12	2672	279	84	1.80	1.45	1.28
14	6783	678	174	1.77	1.47	1.31
16	17270	1683	364	1.74	1.48	1.32
18	41558	4102	751	1.72	1.49	1.34
20	91493	9905	1318	1.69	1.50	1.34
22	175921	22955	2548	1.66	1.50	1.34
24	290082	53039	5733	1.62	1.50	1.36
26	395355	110372	10080	1.58	1.50	1.35
28	463234	202565	22055	1.53	1.49	1.36



$1 + b + \dots + b^d$

$= 24, \ d = 6$

$= 19, \ d = 6$

h_1, h_2

? $\frac{h_1 + h_2}{2}$, $\max(h_1, h_2)$

- Beyond Classical Search

- Substitution Cypher

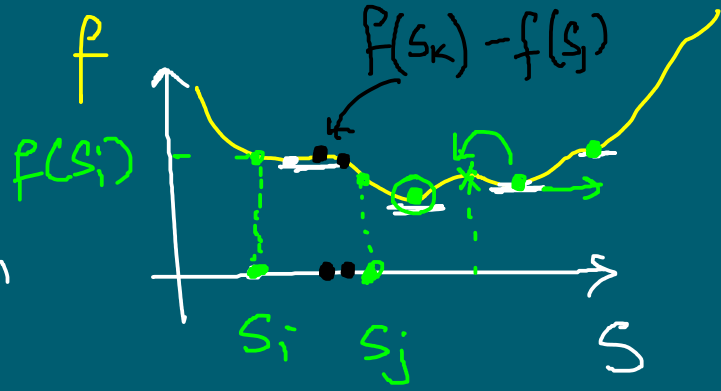
  26 !

  a b . . . z
  ↓ Perm.
  b z . . . c

- Travelling Salesman Problem
  - Trajectory planning
  - VLSI circuit layout
- Knapsack
  - Cargo shipment
  - Portfolio management
  - Sattellite channel allocation
- Scheduling problems

Local Search (Monte Carlo)

- (S, f) - feasible solutions and objective function



min/max

- $\mathcal{N}$ - neighborhood $\qquad \mathcal{N}(s_i)$

1. Given (S, f), initialize with i^{th} state $s_i \in S$
2. Pick jth state from $\mathcal{N}(i)$; $s_j \in S$
3. If $f(j) < f(i)$ then replace current state with j.
4. If $f(j) \geq f(i)$ then check for $\forall j \in \mathcal{N}(i)$ Break
   i-is local minimum
5. Go to step 2.

Defⁿ A state $i^* \in S$ is called a local optimum w.r.t. $\mathcal{N}$ for (S, f) if $f(i^*) \leq f(j)$ $\forall j \in \mathcal{N}(S_{i^*})$.

Metropolis Algorithm

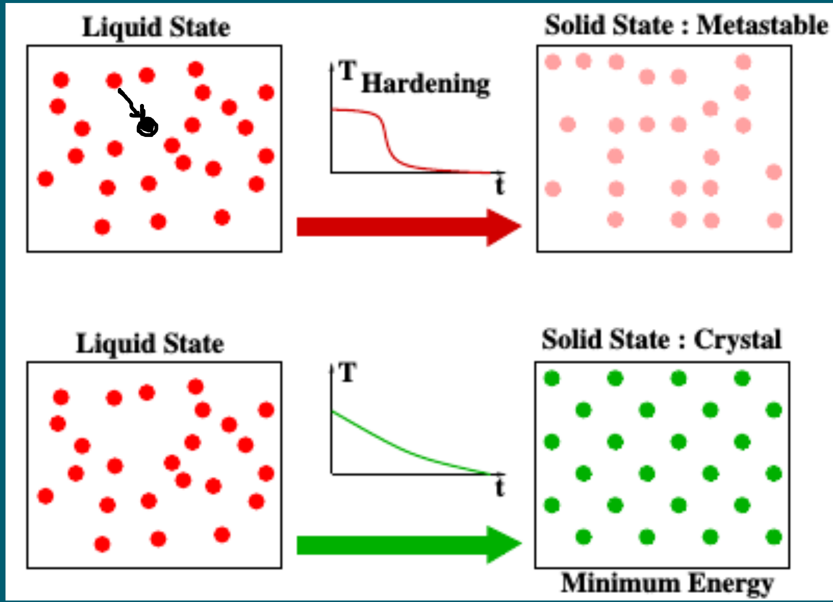
- 1953 – Metropolis, Rosenbluth, Teller
- Simulate "annealing"

$\Rightarrow$ if $f(j) > f(i)$

accept jth state with

(Metropolis acceptance)

$$p_c = \exp\left[-(f(j) - f(i))/c\right]$$



- Solid to a very high temp
  'melting'

- Cool the solid a very particular temperature decreasing scheme in order to reach a solid state of min energy.

$(S, f) - \mathcal{N}$

1. Initial state i

2. $j \in \mathcal{N}(i)$

3. If $f(j) < f(i)$ then $i := j$

   Else $i := j$ with prob $e^{\left(-\frac{(f(j) - f(i))}{c}\right)}$ $\quad (f(j) \geq f(i))$

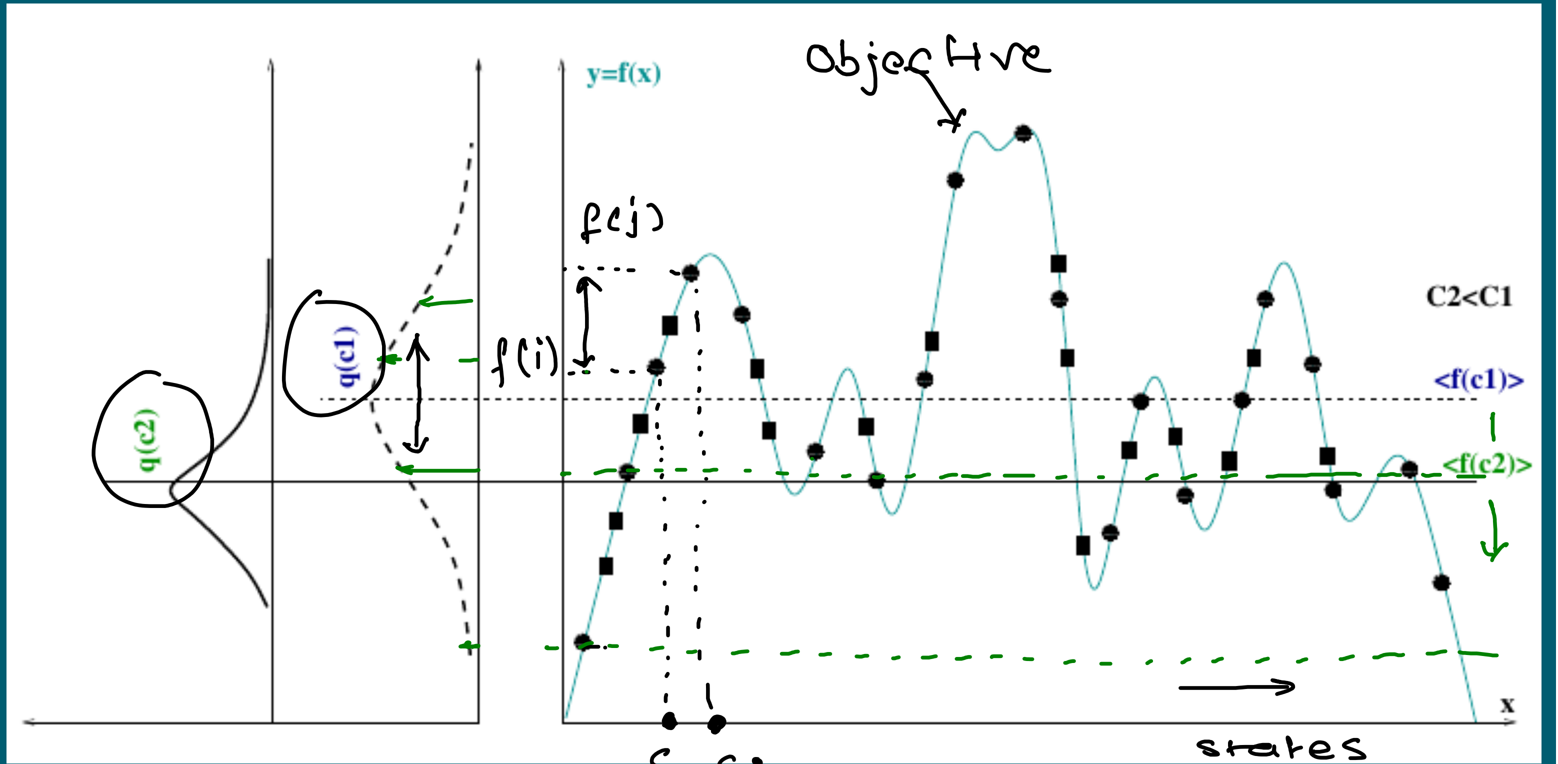
4. Goto step 2. and look for convergence.

Simulated Annealing

- fixed "c"

$P_c(\text{accept } j \mid s_i)$

$$= \begin{cases} 1 & f(j) < f(i) \\ e^{\left(-\frac{[f(j)-f(i)]}{c}\right)} & \text{else} \end{cases}$$



$S_i \; S_j$

$f(s_0) + \ldots + f(s_2) + \ldots + f(s_{N-1})$

$S_0 \to S_1 \to S_2 \ldots \to S_{N-1}$

total = N states visited

- $P_c(x = i) = \frac{1}{N(c)} e^{-f(i)/c} = q(c)$

$N(c) = \sum_{j \in S} e^{-f(j)/c}$

Simulated Annealing

temp Lengthparam

1. Initialize $i = i_{start}$, $k = 0$, $c_k = c_0$, $L_k = L_0$

2. Repeat (on k)

3. For $l = 0$ to L_k do

   - Generate a solution j from $N(s_i)$ of current soln.
   - If $f(j) < f(i)$ then $i = j$ (j becomes current sol)
   - Else j becomes current sol^n. with prob $e^{-\frac{(f(j)-f(i))}{c_k}}$

4. $k = k+1$

5. Compute (L_k, c_k)

6. Until $c_k \simeq 0$.

Knapsack Problem:

$x \in \{0,1\}^n$ – n items – decision (selection) variables

$v \in \mathbb{R}_+^n$ – values associated with each item

$$f(x) = \sum_{i=1}^{n} x_i v_i \qquad \text{Objective function} \quad \text{— A}$$

P – weight limit of Knapsack

$\omega \in \mathbb{R}_+^n$ – weights of items

$$\sum_{i=1}^{n} x_i \omega_i \leq P \qquad \text{— B}$$

$\max f(x)$

s.t. $\sum_{i=1}^{n} \omega_i x_i \leq P$

n	10	20	30	40	50	100
2^n	10^3	10^6	10^9	10^{12}	10^{15}	10^{30}

excess $\Delta = \min\left(0,\ P - \sum_{i=1}^{n} \omega_i x_i\right)$

$(S, f), N$

set of feasible sol^n ✓ ✓ ✓

$$\bar{f}(x) = f(x) + \mu \underbrace{\frac{\Delta}{P}}_{\text{penalty}}$$

k

x_i: | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |

↓

x_j: | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 |

03.02.2022

$PT \xrightarrow{\phi} CT$

$\phi : \{a, \dots, z, ?\} \to \{a, \dots, z, ?\}$

↳ $1.1 = 26!$

n	2^n	n!	2^n	$n!$	ratio $\frac{n!}{2^n}$
10	1.024 10^3	3.628 10^6	1 micro second	3.6 mili seconds	3.6 10^3
20	1.048 10^6	2.432 10^{18}	1 mili second	77 years	2.3 10^{12}
30	1.073 10^9	2.652 10^{32}	1 second	8.4 10^{15} years	2.47 10^{23}
40	1.099 10^{12}	8.159 10^{47}	18 minutes	2.5 10^{31} years	7.4 10^{35}
50	1.125 10^{15}	3.041 10^{64}	13 days	9.6 10^{47} years	2.7 10^{49}
60	1.152 10^{18}	8.320 10^{81}	36 years	2.6 10^{47} years	7.2 10^{63}
70	1.180 10^{21}	1.197 10^{100}	37 10^3 years	3.8 10^{83} years	1 10^{79}
80	1.208 10^{24}	7.156 10^{118}	38 10^6 years	2.2 10^{102} years	5.9 10^{94}
90	1.237 10^{27}	1.485 10^{138}	39 10^9 years	4.7 10^{121} years	1.2 10^{111}
100	1.267 10^{30}	9.332 10^{157}	40 10^{12} years	2.9 10^{141} years	7.3 10^{127}

One evaluation 10^{-9} seconds

Substitution cypher:

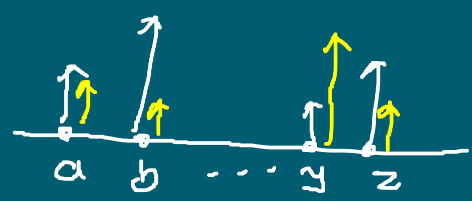
(S, f) $\underline{N}$

set of perm on alphabets $26!$

Codedtext	Plaintext
QAEW QA Z LZNG ICABWN'J	ONCE ON A DARK WINTER'S
LZV, IYWA BYW VWRRQI KQT	DAY, WHEN THE YELLOW FOG
YOAT JQ BYCEG ZAL YWZPV CA	HUNG SO THICK AND HEAVY IN
BYW JBNWWBJ QK RQALQA	THE STREETS OF LONDON
BYZB BYW RZDMJ IWNW	THAT THE LAMPS WERE
RCTYBWL ZAL BYW JYQM	LIGHTED AND THE SHOP
ICALQIJ SRZFWL ICBY TZJ ZJ	WINDOWS BLAZED WITH GAS
BYWV LQ ZB ACTYB, ZA	AS THEY DO AT NIGHT, AN
QLL-RQQGCAT RCBBRW TCNR	ODD-LOOKING LITTLE GIRL SAT
JZB CA Z EZS ICBY YWN	IN A CAB WITH HER FATHER
KZBYWN ZAL IZJ LNCPWA	AND WAS DRIVEN RATHER
NZBYWN JRQIRV BYNQOTY	SLOWLY THROUGH THE BIG
BYW SCT BYQNQOTYKZNWJ.	THOROUGHFARES.

$$\begin{array}{lcccc} & x_1 & x_2 & x_3 & x_4 \\ P_c(x) = & 0.5 & 0.3 & 0.1 & 0.1 \\ P_{M_0}(x) : & 0.1 & 0.1 & 0.3 & 0.5 \end{array}$$

$P_c(x)$ $\quad$ $P_{M_0}(x)$



$x \in \mathcal{A}$, $P_{M_0}(x)$

first order LM

$P_{M_1}(x_{i+1} \mid x_i)$ $\quad$ P_{M_1}

Perm:

$$\begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \\ \downarrow & \downarrow & \downarrow & \downarrow \\ x_4 & x_3 & x_1 & x_2 \end{array}$$

$P_{M_1}(x_{i+1} \mid x_i)$ Language model

Coded Text $= t_1 t_2 t_3 t_4 \ldots, t_K$

$$P(CT \mid M_1) = P(t_1 t_2 \ldots t_K \mid M_1)$$

Plausibility $f(\phi = I) = \underline{P_{M_1}(t_1)} \cdot P_{M_1}(t_2 \mid t_1) \ldots P_{M_1}(t_K \mid t_{K-1})$

Objective function

$$\overset{I}{CT_1} \xrightarrow{\Phi_1} CT_2 \xrightarrow{\Phi_2} CT_3 \ldots \xrightarrow{\Phi_{n-1}} CT_n = PT$$

$$f_1 \qquad f_2 \qquad f_3 \qquad \hat{f}_n$$

Max.

$$f(\phi) = \sum_{i=1}^{K} \log\left(P_{M_1}(\phi(t_2) \mid \phi(t_1))\right) + \log P_{M_1}(\phi(t_1))$$

$$I \longrightarrow x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5 \quad x_6$$

$$\left.\begin{array}{l} \phi_i \longrightarrow x_2 \quad \boxed{x_1} \quad x_4 \quad x_5 \quad \boxed{x_6} \quad x_3 \\ \quad \downarrow \mathcal{N} \\ \phi_{i+1} \longrightarrow x_2 \quad x_6 \quad x_4 \quad x_5 \quad x_1 \quad x_3 \end{array}\right\} \text{transposition}$$

$$\phi_{i+1} \in \mathcal{N}(\phi_i)$$

$$P(\phi_{i+1} \text{ accept}) = \begin{cases} e^{-\frac{(f(\phi_{i+1}) - f(\phi_i))}{c}} & f(\phi_{i+1}) \leq f(\phi_i) \\ 1 & f(\phi_{i+1}) > f(\phi_i) \end{cases}$$

$$e^{-\left[\frac{\log(Pl(\phi_{i+1})) - \log(Pl(\phi_i))}{c}\right]}$$

$$\log\left(\frac{Pl(\phi_{i+1})}{Pl(\phi_i)}\right)^{-1} / c$$

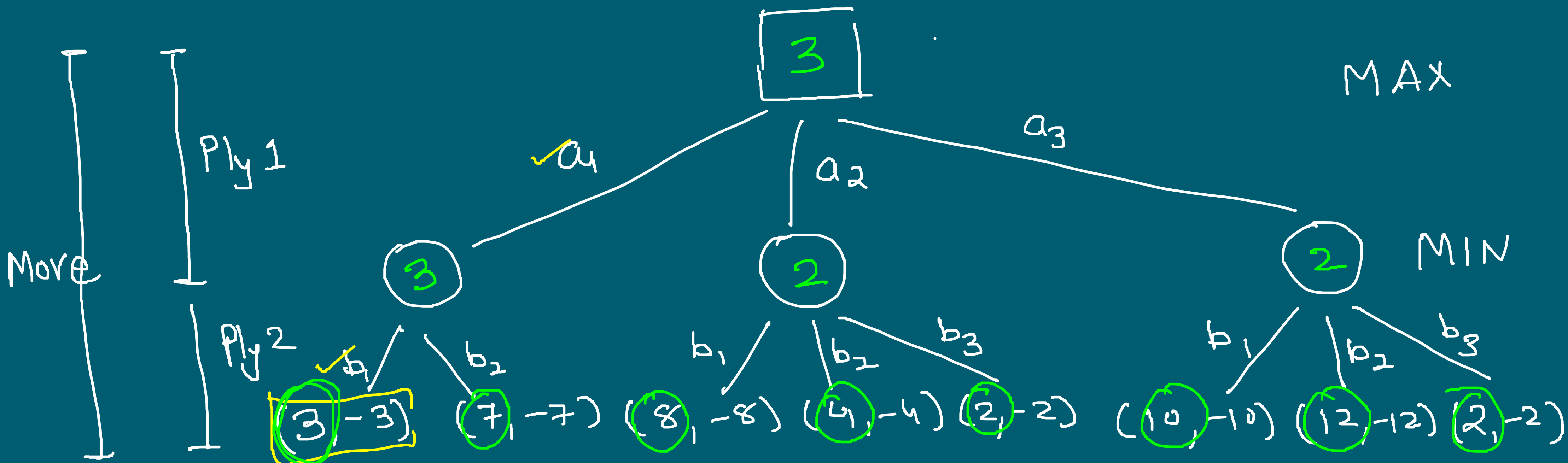
$$\approx e \qquad = Pl(\phi_i) / Pl(\phi_{i+1}) + \mathbb{C}$$

MCMC — Markov Chain Monte Carlo . 20th Cent

[Diaconis]

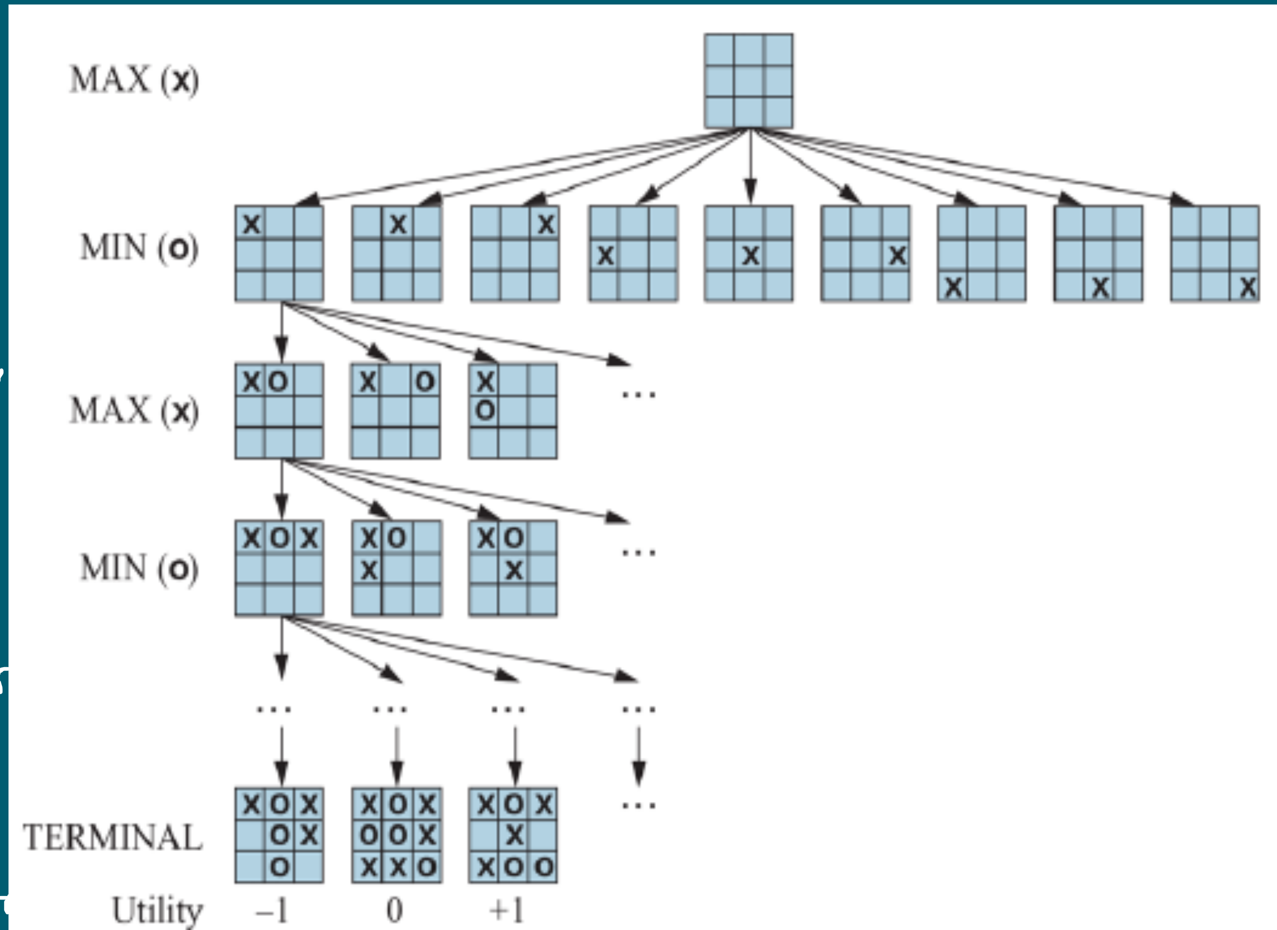
Adversarial Search.

- Zero sum game
- Perfect Info (fully observable)



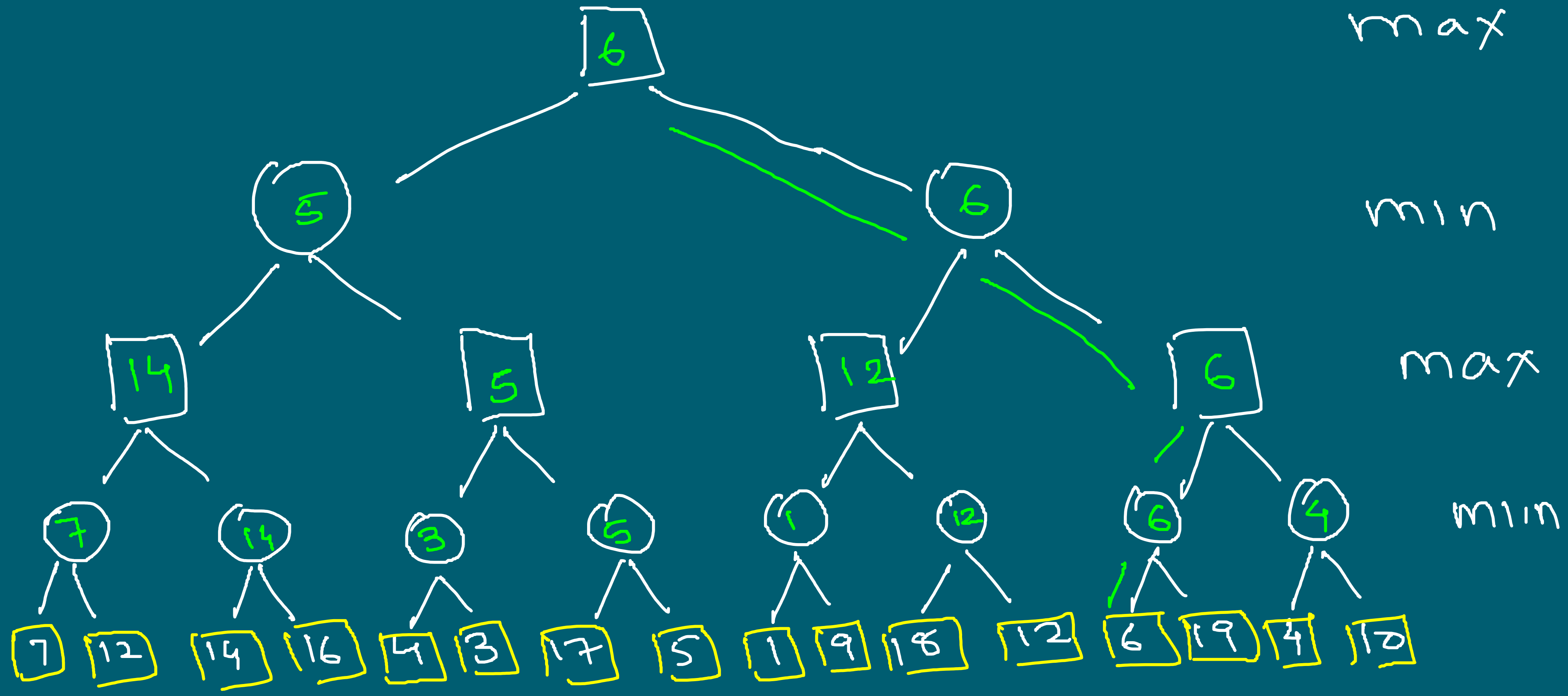
$$\max_{a_1, a_2, a_3} \left\{ \min_{b_i} \left(U_{til_i} \right) \right\} \sim \max\big(\min(3,7), \min(6,4,2), \min(10,12,2)\big)$$

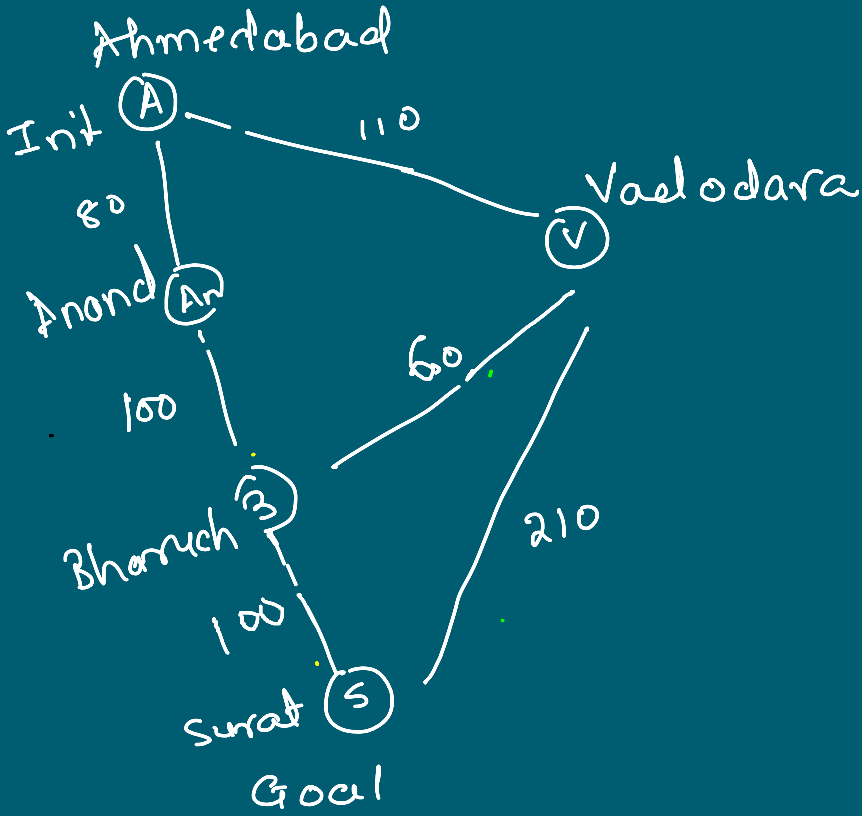
- MINIMAX agent

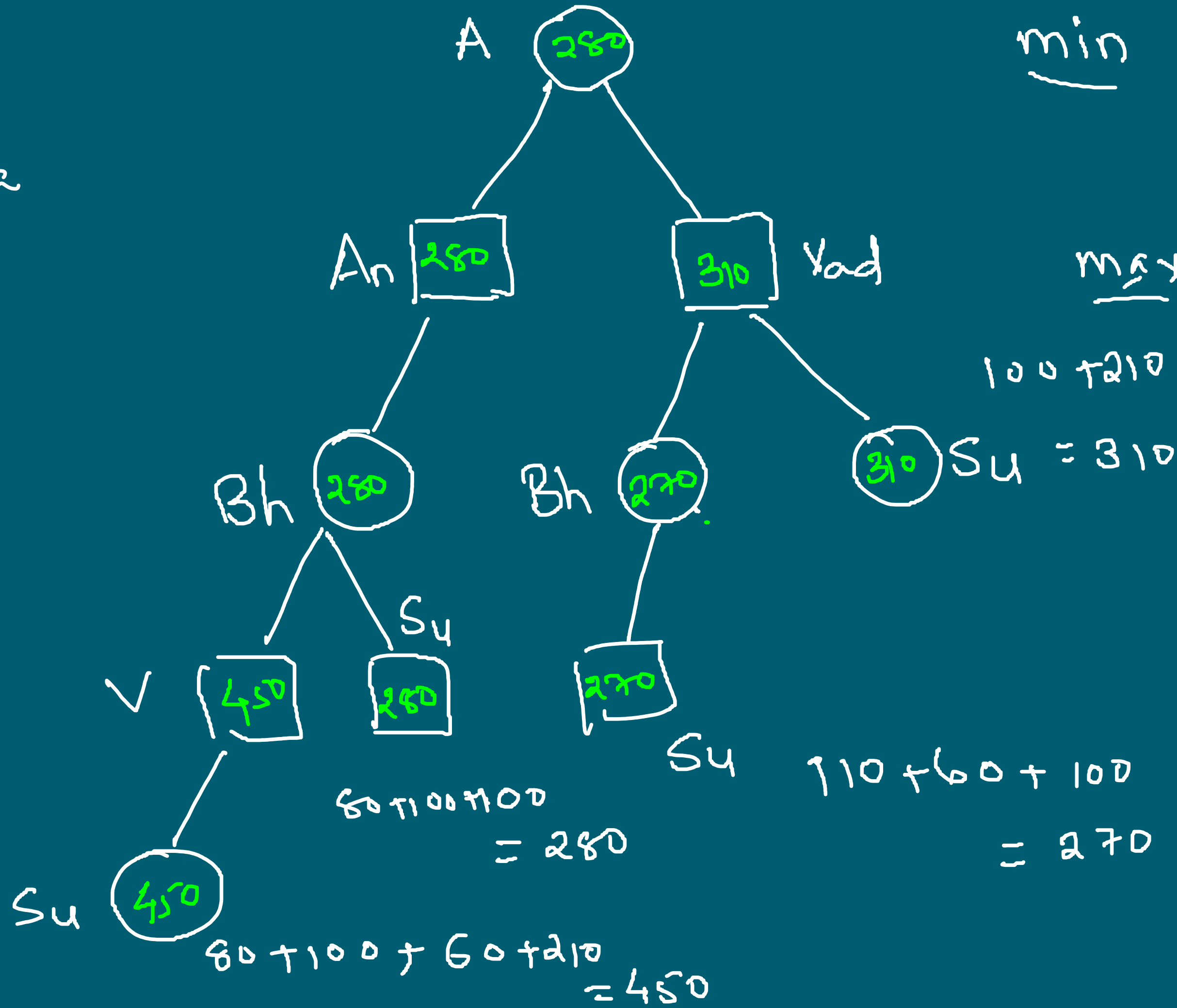


$$\text{MINIMAX}(s) = \begin{cases} \text{Utility}(s, \max) & s \text{ is term/leaf} \\ \min_{a \in A(s)} \big(\text{MINIMAX}(\text{Re}(s,a))\big) & \text{minplayer} \\ \max_{a \in A(s)} \big(\text{MINIMAX}(\text{Re}(s,a))\big) & \text{maxplayer} \end{cases}$$

Ex.

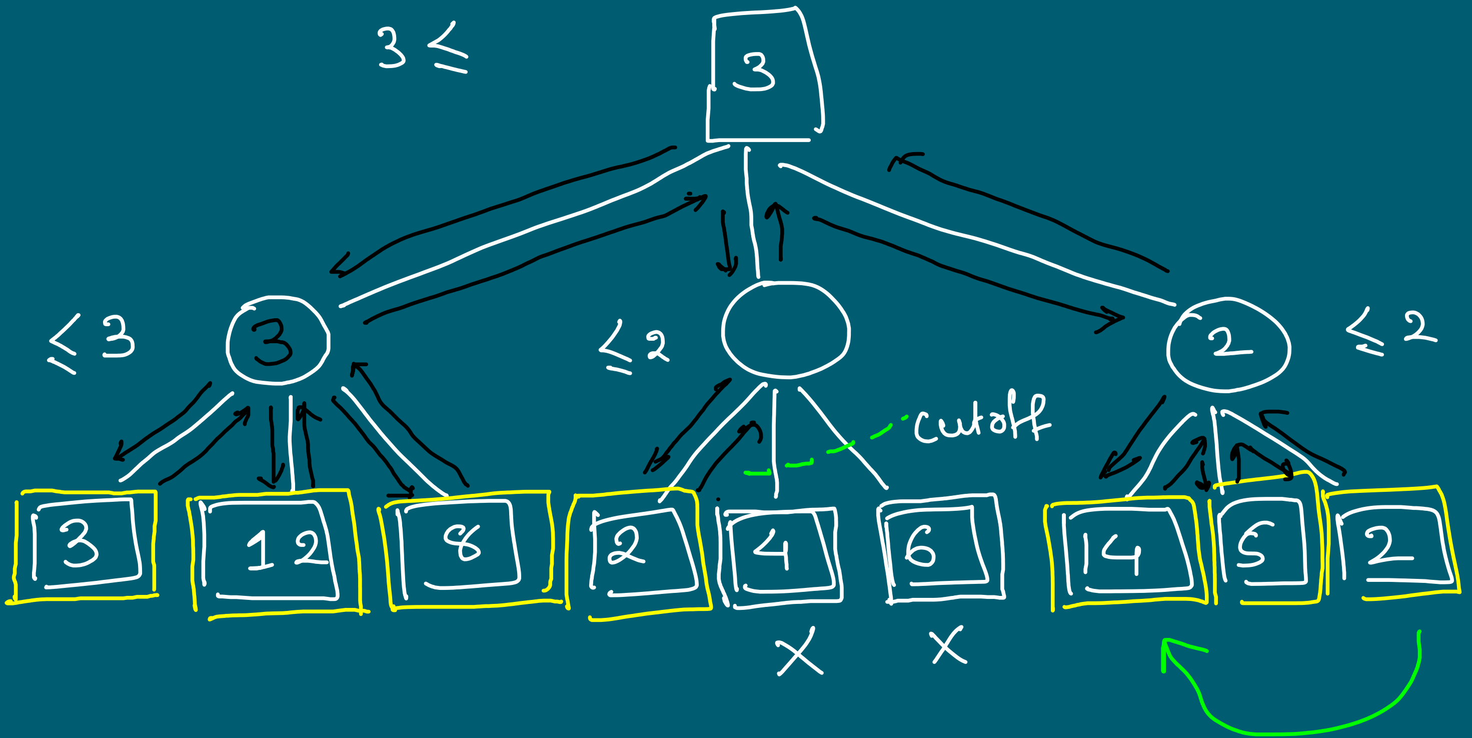

6
5
6
14
5
12
6
7
14
3
5
1
12
6
4
7
12
14
16
4
3
17
5
1
9
18
12
6
19
4
10
max
min
max
min


Ahmedabad
Init
A
110
80
Vadodara
V
Anand
An
60
100
Bharuch
3
210
100
Surat
S
Goal


A
280
An
280
310
Vad
Bh
280
Bh
270
310
Su = 310
V
450
Su
280
270
Su
Su
450
80+100+100
= 280
80+100+60+210
= 450
110+60+100
= 270
100+210

min

max

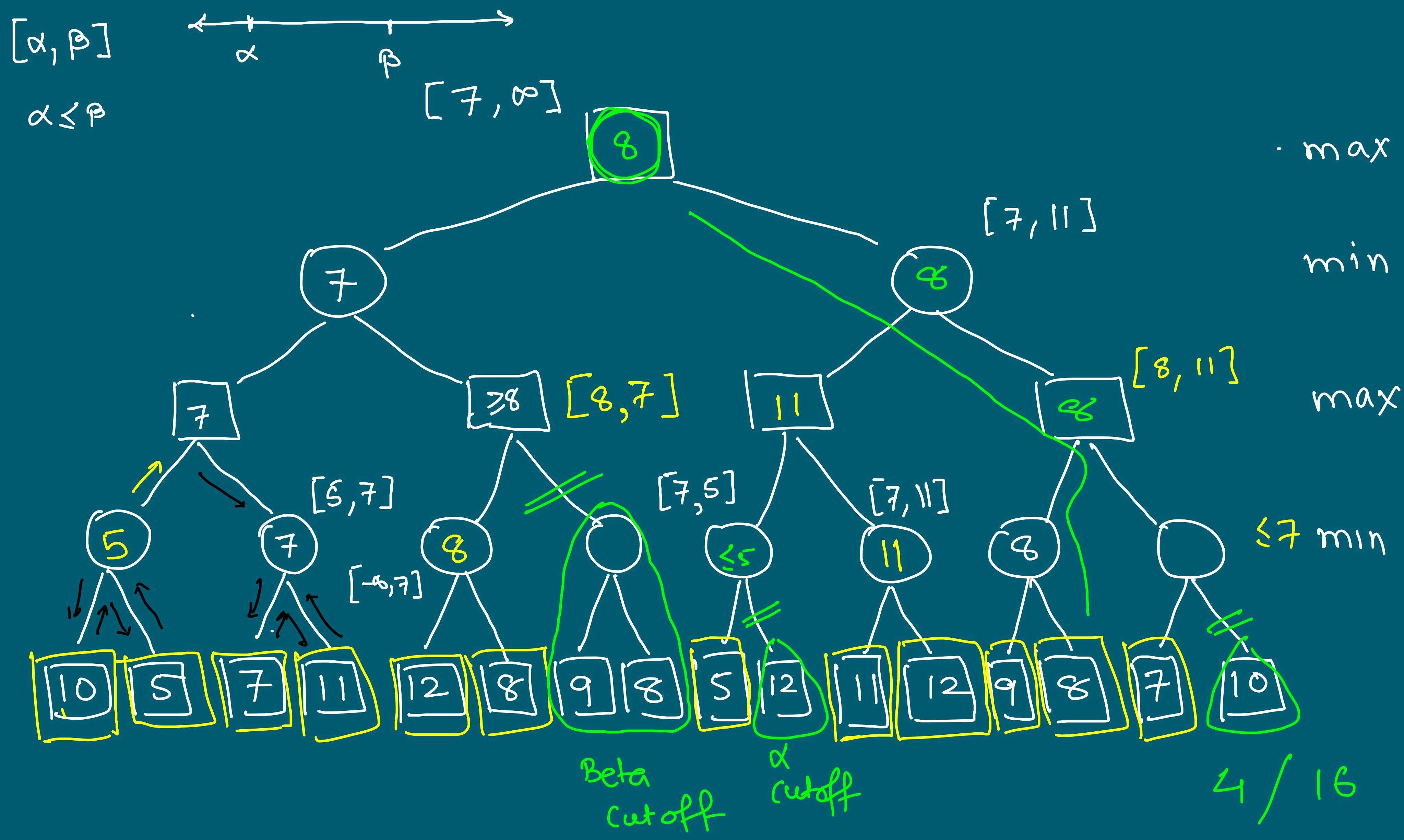


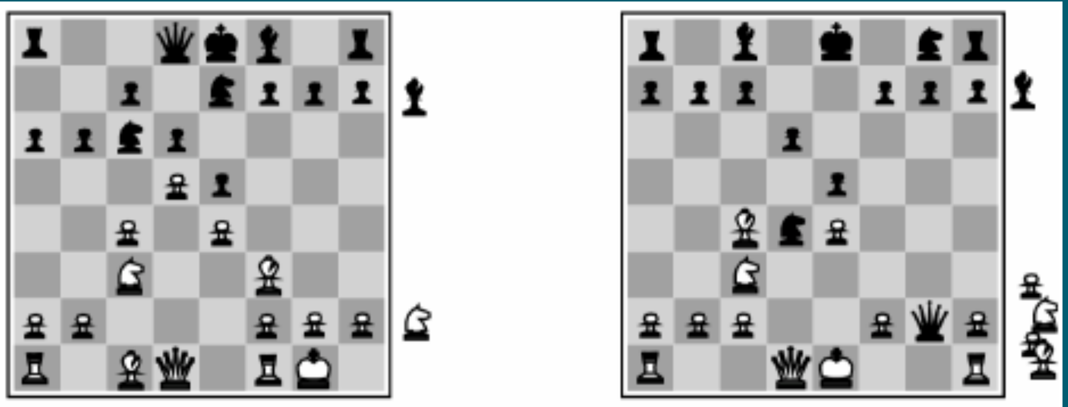
$O(b^m)$

$\downarrow$

$O(b^{m/2})$

α-β pruning


[α, β]
α ≤ β
α
β
[7, ∞]
8
max
[7, 11]
7
8
min
7
≥8
[8, 7]
11
8
[8, 11]
max
5
[5, 7]
7
[-∞, 7]
8
[7, 5]
≤5
[7, 11]
11
8
≤7 min
10
5
7
11
12
8
9
8
5
12
11
12
9
8
7
10
Beta cutoff
α cutoff
4 / 16



Black to move
White slightly better

White to move
Black winning

$$u(x) = \omega_1 f_1(x) + \ldots + \omega_n f_n(x).$$

$f_1(x)$ = #Wpawns − #blackpa.

⋮

$f_i(x)$ = #wknights − #bknights

Until game is over

1. Start with the current position as a MAX
2. Expand the game tree a fixed # of ply
3. Apply the evalution/utility function to each leaf node
4. Back-up values bottom-up (MINIMAX).
5. Pick the move assigned to MAX at root.
6. Wait for opponent to respond

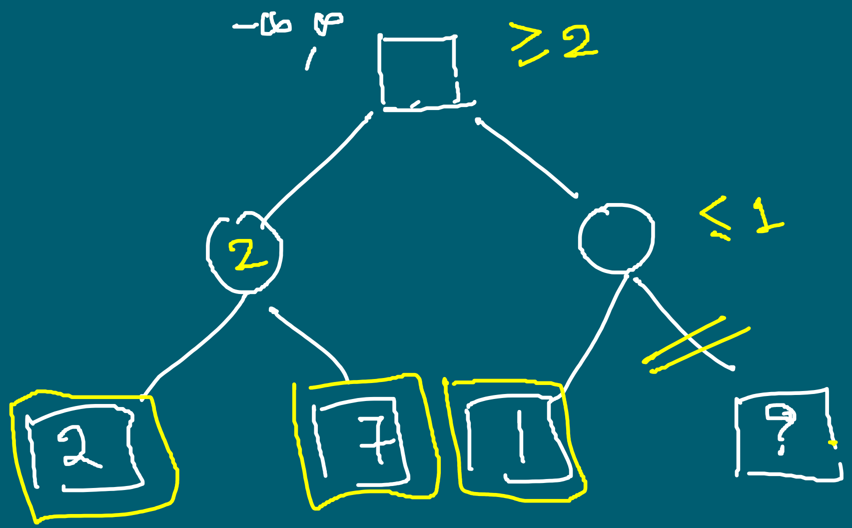
MINIMAX · [DFS] Time $O(b^m)$ Space $O(bm)$

α-β $O(b^{m/2})$

Best case · Knuth

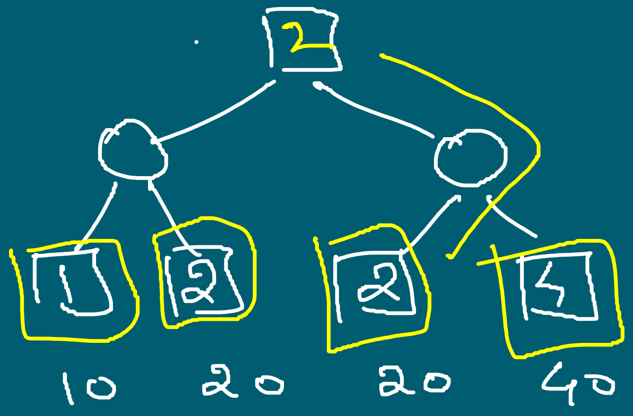
Alpha-Beta Pruning

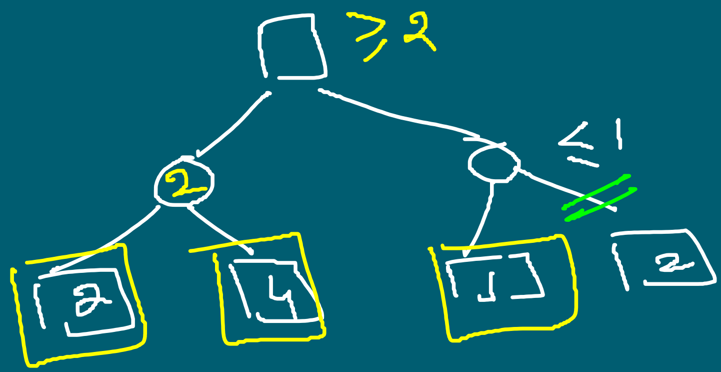
· "If you have an idea which is surely bad, don't take the time to see how truly awful it is!" — Pat Winston

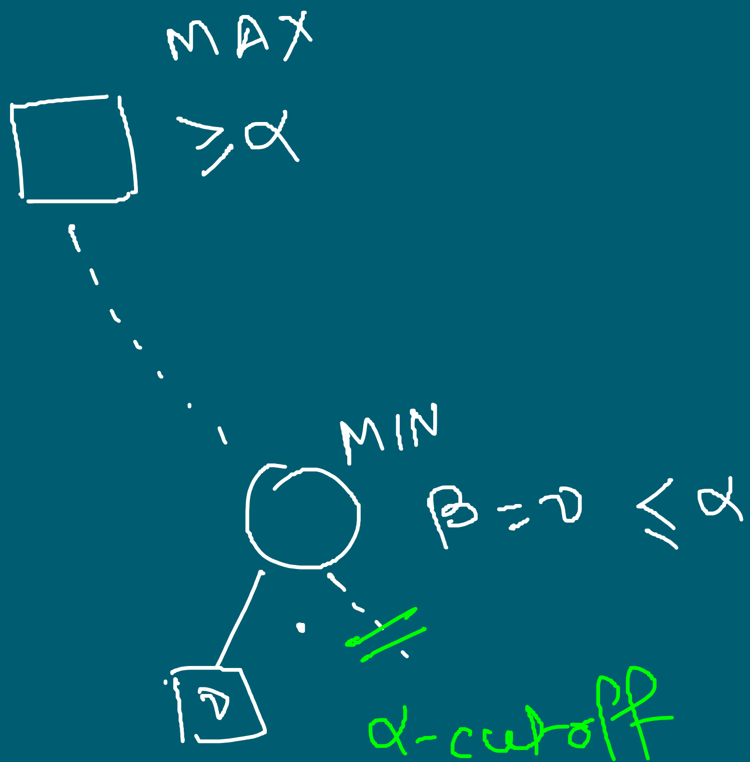


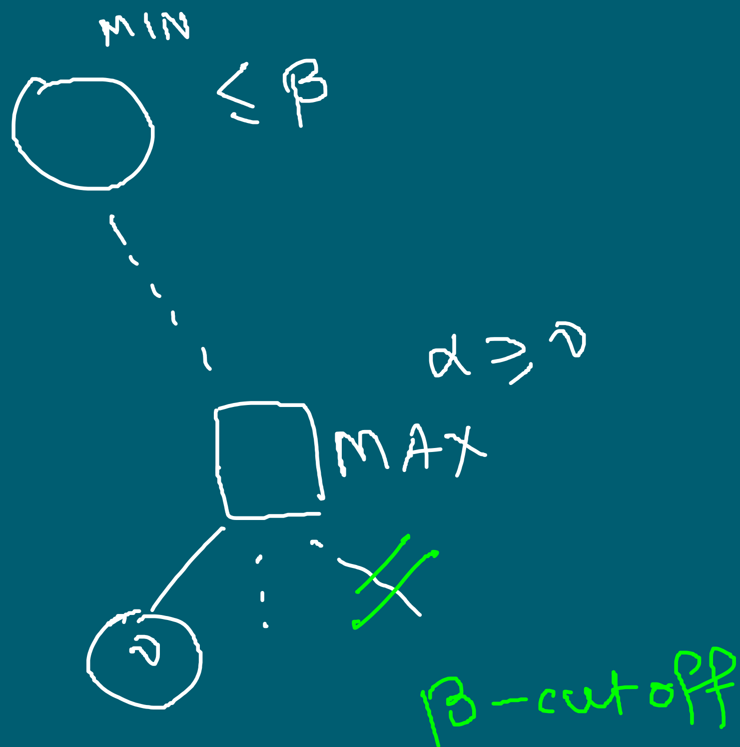
· α – MAX's current LB
· β – MIN's current UP
· $\alpha \geq \beta$ pruning
· Pass α, β down to child
· Update α, β during search
  · MAX — updates α
  MIN — updates β

not the value but order!









- 1997 – Deep Blue
  - 200 million/sec

$b \sim 35$

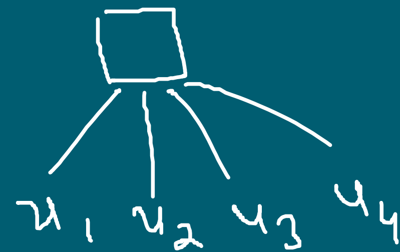
$m \sim 40$

$\sim 35^{40}$

+ Lookup

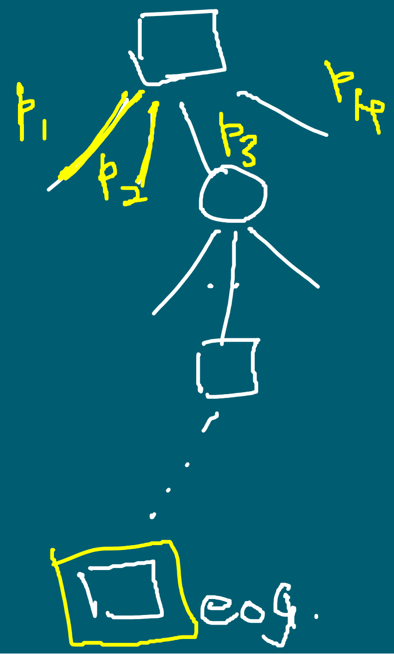
- Othello
- Go 2006 → DeepMind + Go

Smallest utility first



Largest utility first

Monte Carlo Tree Search



Probabilistic Inference

Ex. 1 $X : \{1, 2, 3, 4, 5, 6\}$ $Y : \{1, 2, 3, 4, 5, 6\}$

$P_X : \{0.1, 0.2, 0.1, 0.2, 0.4, 0\}$ $P_Y : \{0.3, 0.1, 0.2, 0.3, 0.1, 0\}$

$Z = X + Y$ Condition $Z = 5$ Distri / Mass fun Infer about X ? $\boxed{\text{Query}}$

$$P(X=1, Y=1) = P(X=1) \cdot P(Y=1)$$
$$= 0.1 \times 0.3 = 0.03$$

$P(x,y)$	1	2	3	4	5	6
1	0.03					
2						
3						
4						
5						
6						

$$P(Z=5) = 0.1 \times 0.3 + 0.2 \times 0.1 +$$
$$0.1 \times 0.1 + 0.2 \times 0.3$$

$P(X \mid Z=5)$ — $P(X=1 \mid Z=5)$ $P(X=3 \mid Z=5)$ $P(X=5 \mid Z=5$

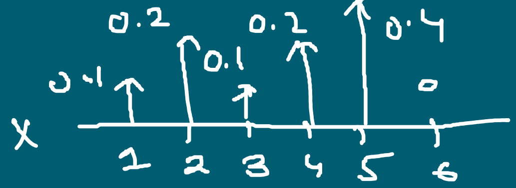
Conditional $P(X=2 \mid Z=5)$ $P(X=4 \mid Z=5)$ $P(X=6 \mid Z=5$

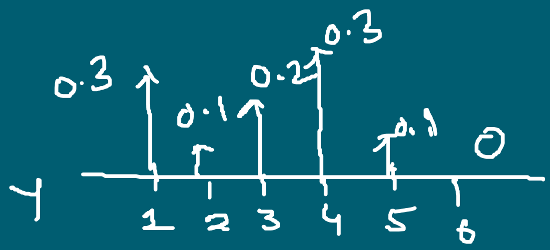
$\boxed{\text{Prob Prog Lang. – WebPPL}}$

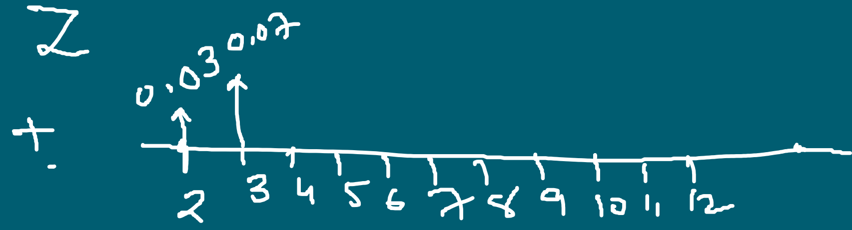
$P_X \, , \, P_Y \longrightarrow P_Z$

$Z = X + Y$

X & Y are independent







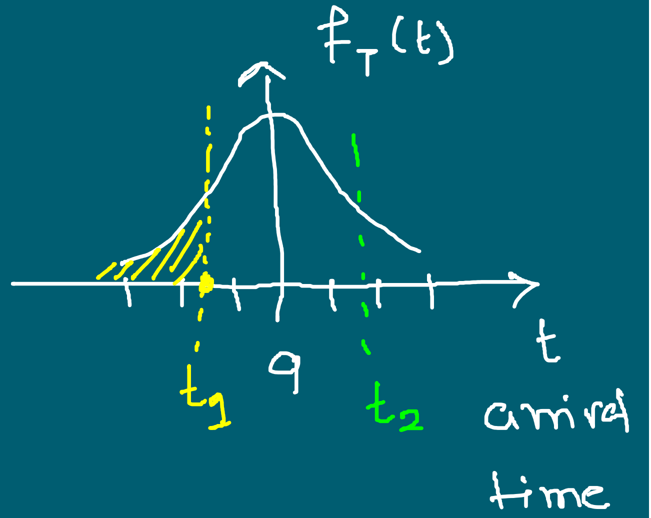
$P_Z = P_X * P_Y$

$P_X(s) = 0.1\,s + 0.2\,s^2 + 0.1\,s^3 + 0.2\,s^4 + 0.4\,s^5 + 0.s^6$

$P_Y(s) = 0.3\,s + 0.1\,s^2 + 0.3\,s^3 + 0.3\,s^4 + 0.1\,s^5 + 0.s^6$

$P_Z(s) = P_X(s) \times P_Y(s) = 0.03\,s^2 + 0.07\,s^3 + \quad s^4 + \quad s^5 + \dots$

$\underline{\text{Ex } 2}$ Normal $(9, 0.5)$ arrival time



$P(\text{meet me at } t) = 0.9$

$f_T(t)$ — Prob Density fun

$P(T \leq t_1)$

$F_T(t) = P(T \leq t)$ Cumulative Distribution fun

$$= \int_{-\infty}^{t} f_T(s)\, ds$$

- R
- matlab
- Python