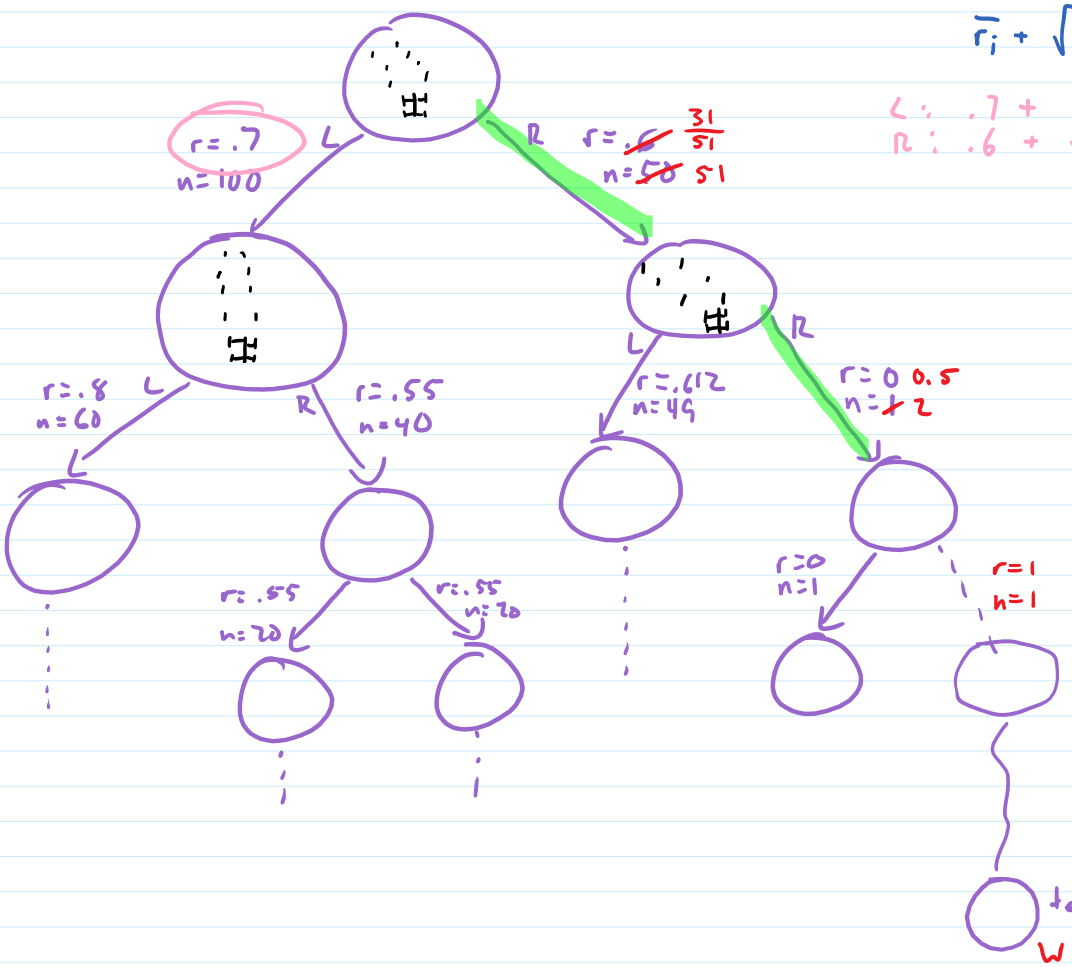
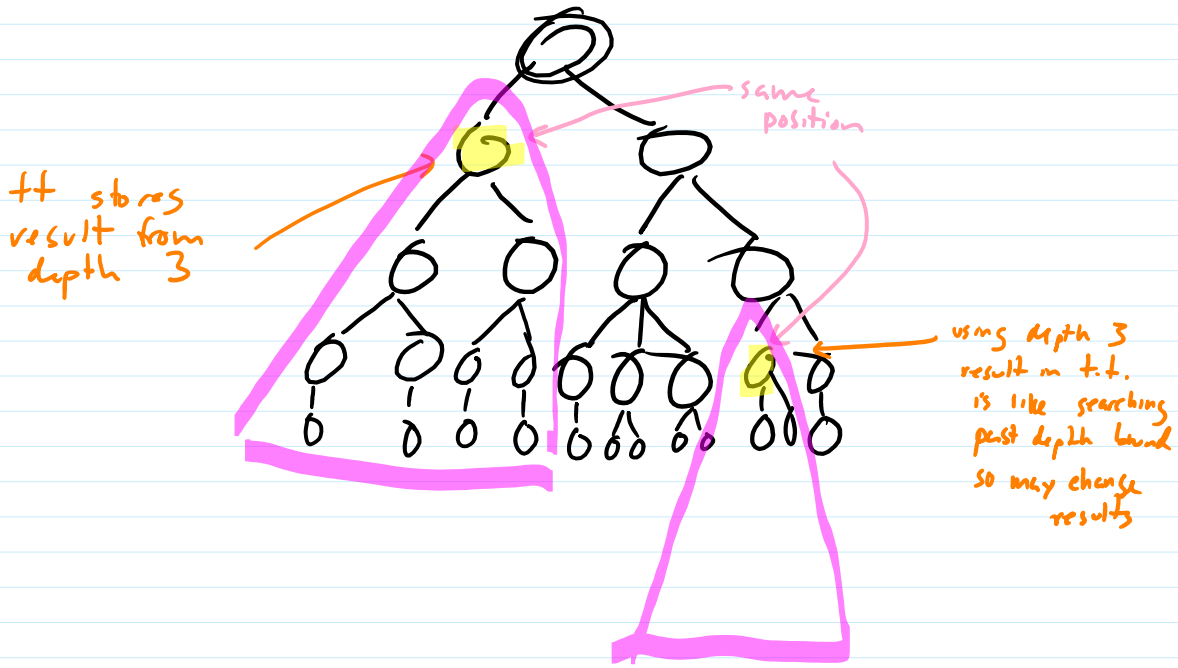

$\bar{r}_i + \sqrt{\frac{2 \ln T}{n_i}}$
L: .7 + √.1002 1.016
R: .6 + √.2004 1.047
r=.7
n=100
L
R
r=.6 31/51
n=50 51
r=.8
n=60
L
R
r=.55
n=40
r=.612
n=49
L
R
r=0 0.5
n=1 2
r=.55
n=20
r=.55
n=20
r=0
n=1
r=1
n=1
terminal pos
WIN!
(a miracle!)
propagate results up tree


same position
tt stores result from depth 3
using depth 3 result in t.t. is like searching past depth bound so may change results

Monte Carlo Tree Search

default policy: random

move-averaged sampling technique (MAST)
maintain reward for actions taken during playout
bias random moves towards higher reward

	n	r̄
capture	100	.60
move again	110	.53
other	240	.40

	n	r̄
100006	~~100000~~ 90004	~~.85~~ ~~.42~~
L		
R		

current playout
LLLRRLRRLL wins

predicate average (PAST)
establish predicates
maintain rewards for moves satisfying predicates
bias towards moves to pos that satisfy preds w/ higher rewards

P_1: A has more in store
P_2: B has more in store
P_3: A has no pit w/ >2 seeds
P_4: $\sim P_3$
P_5: A has empty pits
P_6: $\sim P_5$

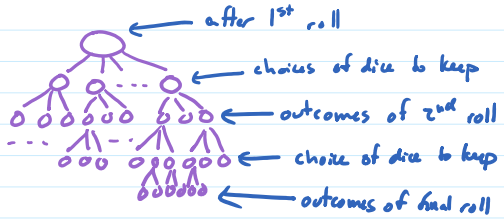
n	r̄	
100^{+1}	0.53	53/101
50	0.4	
50	0.34	
100^{+1}	0.55	56/101
50^{+1}	0.7	36/51
100	0.5	

P1 satisfied
P4 satisfied
P5 satisfied

$\frac{.52+.55+.7}{3} = \sim$

leads to W

stochastic games:



tree policy: UCB at choice nodes
random at random event nodes

imperfect information games:
determinization: pick values for unknown info

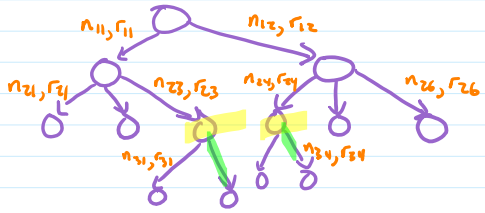
MCTS tree data → playability of game?

variance of score
spread (max-min)
⋮

not too hard / easy
not unbalanced
not drawish
not too long or short

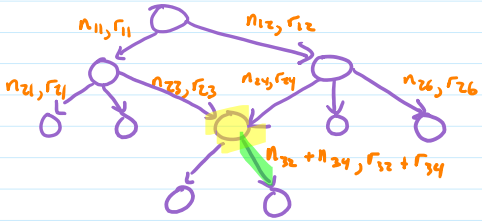
MCDS?

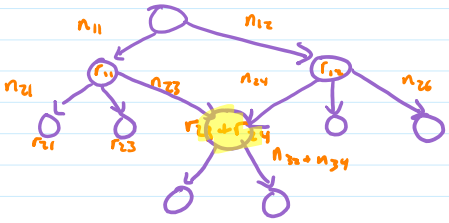
directed acyclic graph (DAG)

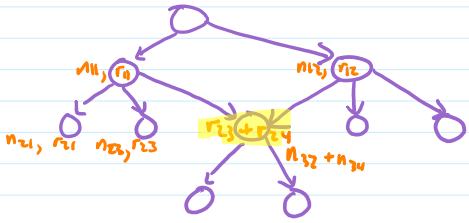


n_{ij} = times move played
r_{ij} = total reward after move

same position after 2 seq of moves







store graph as a map
position → statistics
edges are implicit (given by legal-moves + result)

Two-player games

$$\overline{r_j} \pm \sqrt{\frac{2 \cdot \ln T}{n_j}}$$

if r_j always reward for PI

@ PII nodes, modify form