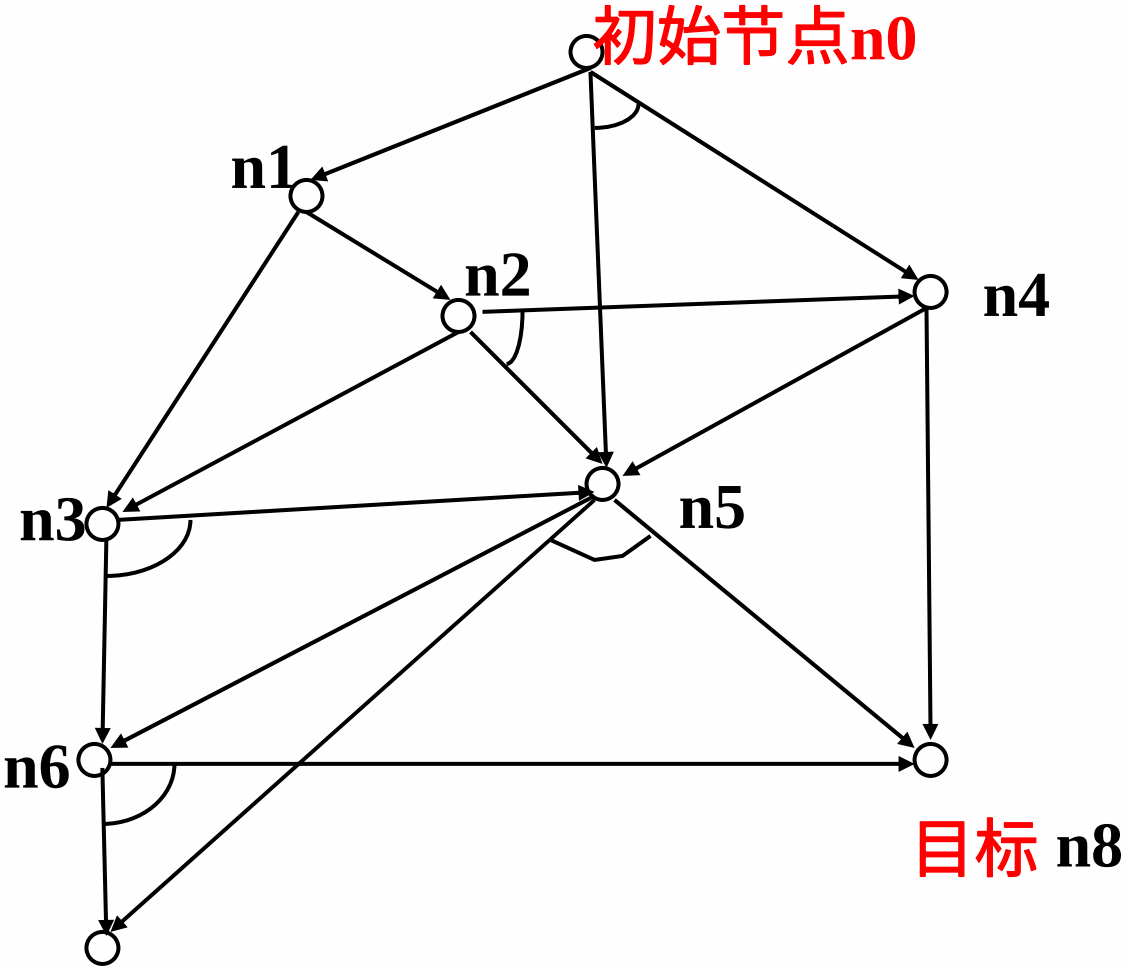
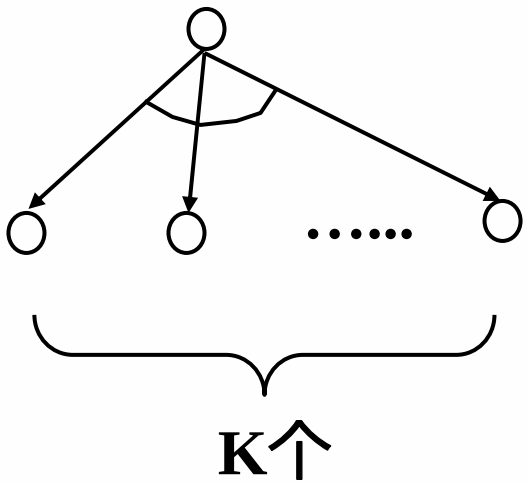
第二章 与或图搜索问题



2.1 基本概念

- 与或图是一个超图，节点间通过连接符连接。
- K-连接符：

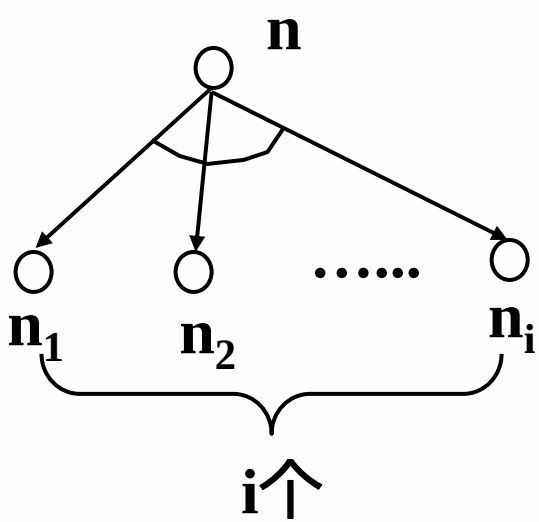


耗散值的计算

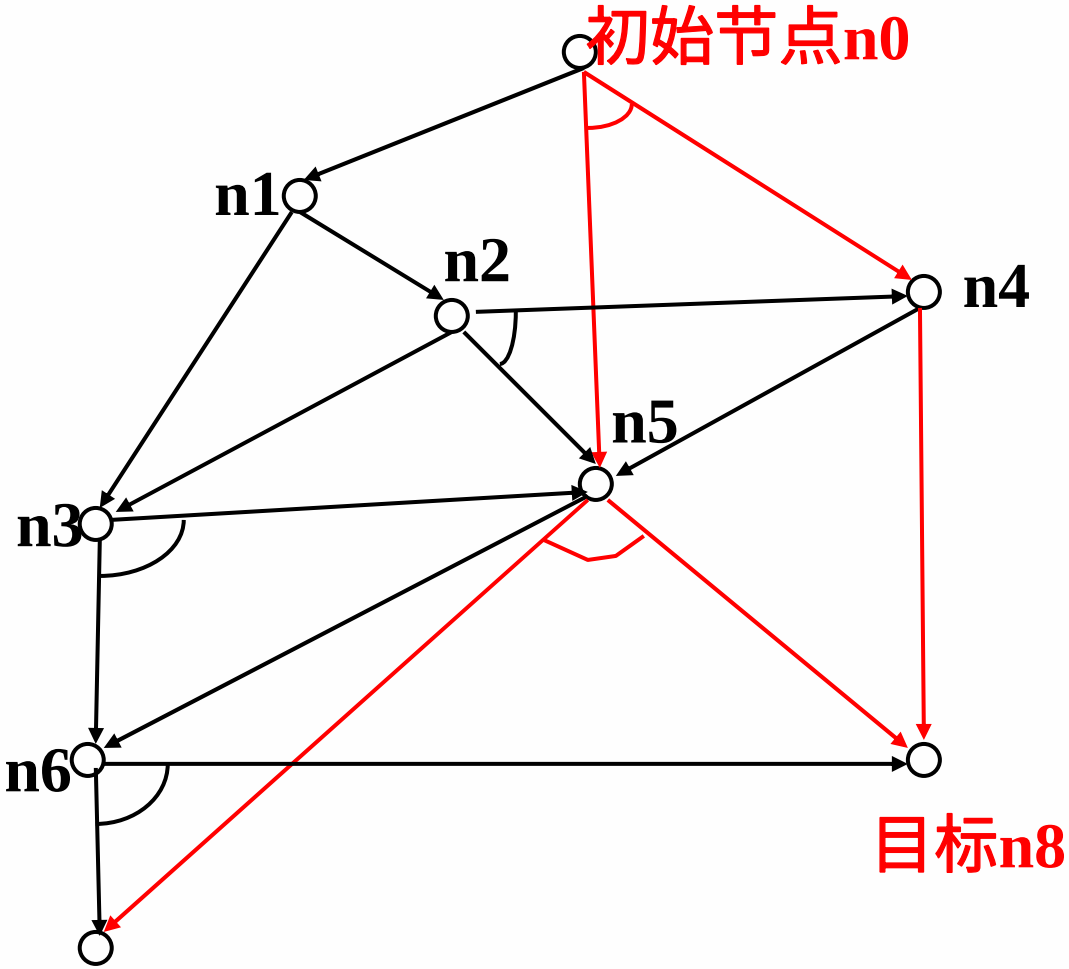
$k(n, N) = C_n + k(n_1, N) + \ldots + k(n_i, N)$

其中：N为终节点集

C_n为连接符的耗散值



• 解图1：



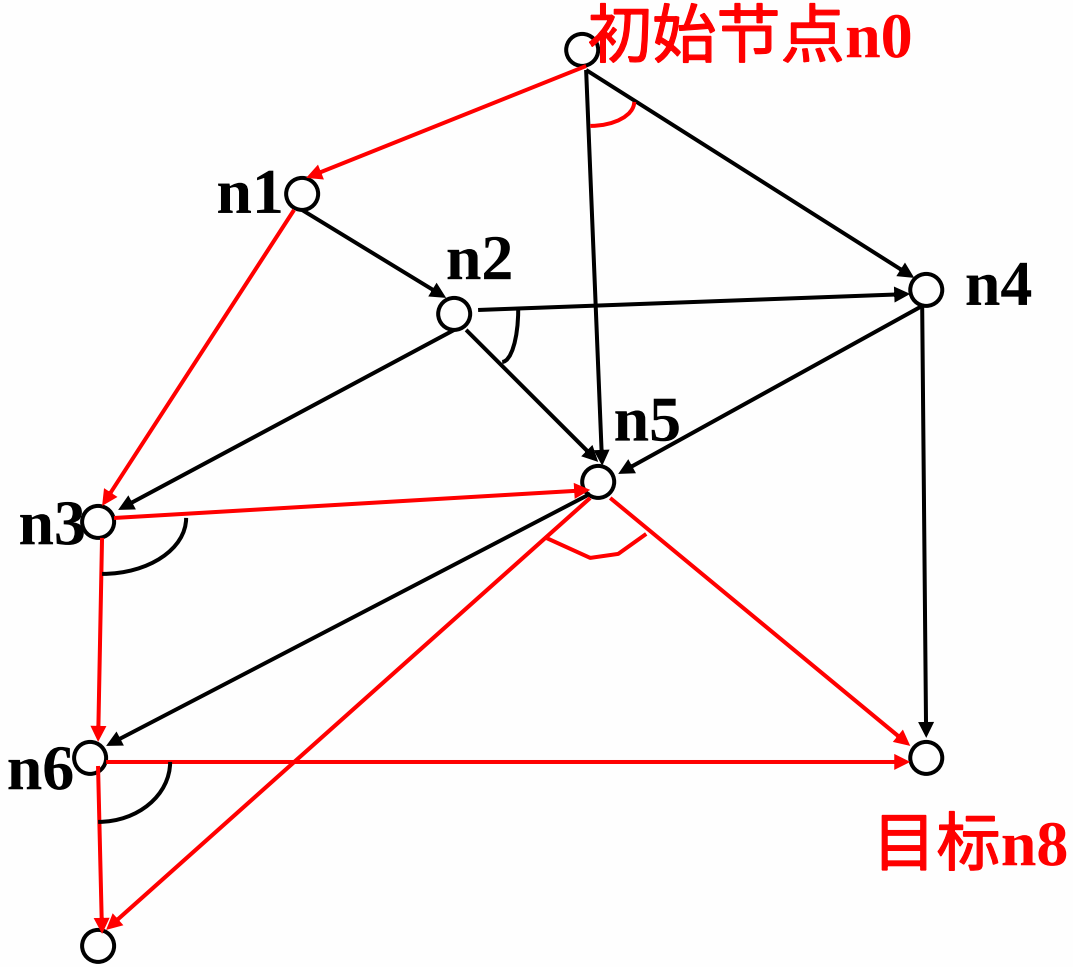
K(n0,N)

=2+k(n4,N)+k(n5,N)

=2+1+k(n8,N)+2+ k(n7,N)+k(n8,N)

=5

• 解图2：



K(n0,N)

=1+k(n1,N)

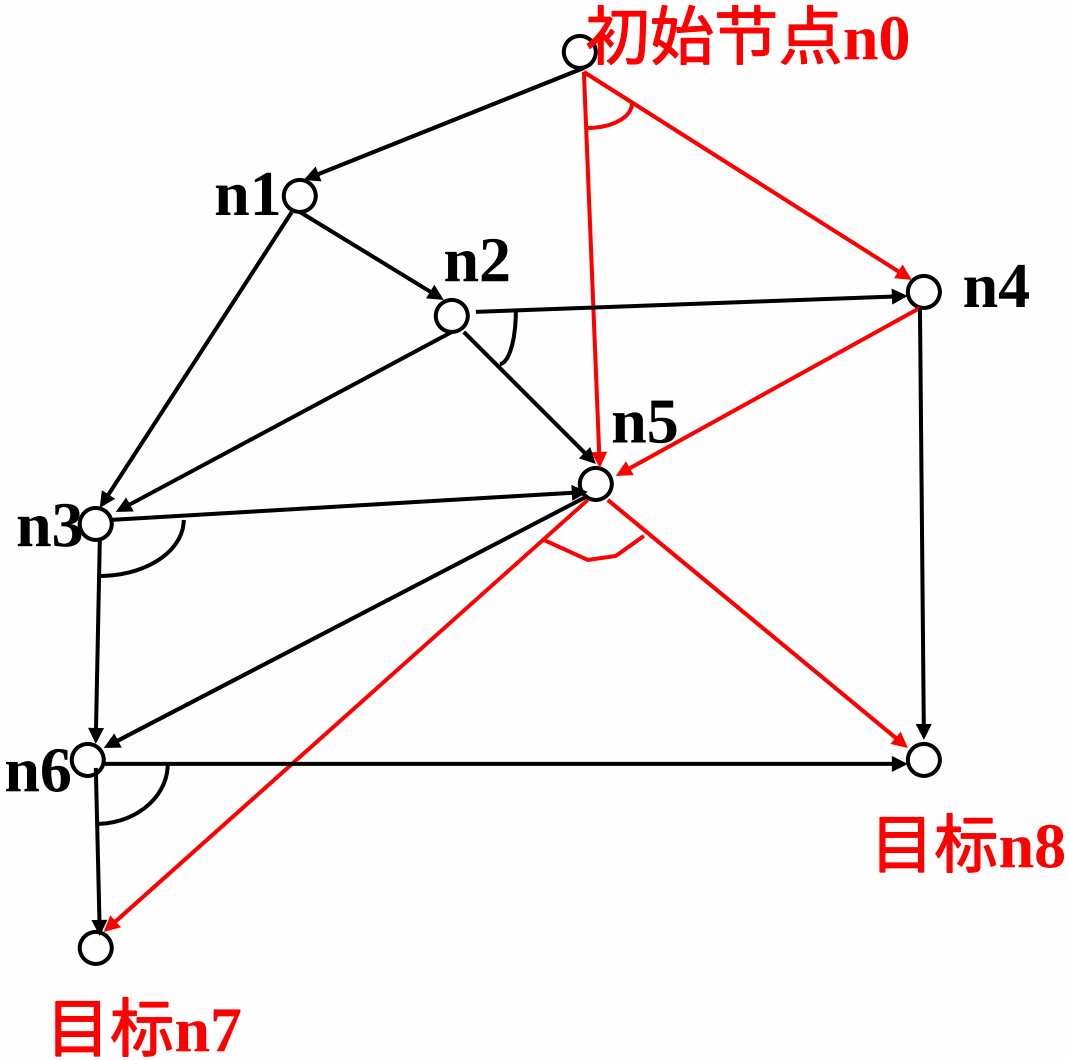
=1+1+k(n3,N)

=2+2+k(n5,N)+k(n6,N)

=4+2+k(n7,N)+k(n8,N)
+2+ k(n7,N)+k(n8,N)

=8

• 解图：



K(n0,N)

=2+k(n4,N)+k(n5,N)

=3+2k(n5,N)

=3+2×2

=7

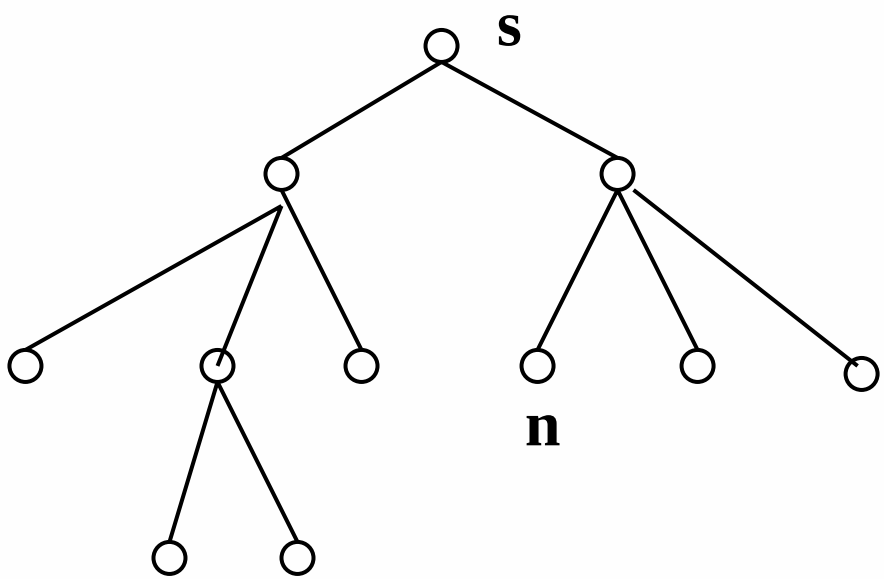
能解节点

- 终节点是能解节点
- 若非终节点有“或”子节点时，当且仅当其子节点至少有一能解时，该非终节点才能解。
- 若非终节点有“与”子节点时，当且仅当其子节点均能解时，该非终节点才能解。

不能解节点

- 没有后裔的非终节点是不能解节点。
- 若非终节点有“或”子节点，当且仅当所有子节点均不能解时，该非终节点才不能解。
- 若非终节点有“与”子节点时，当至少有一个子节点不能解时，该非终节点才不能解。

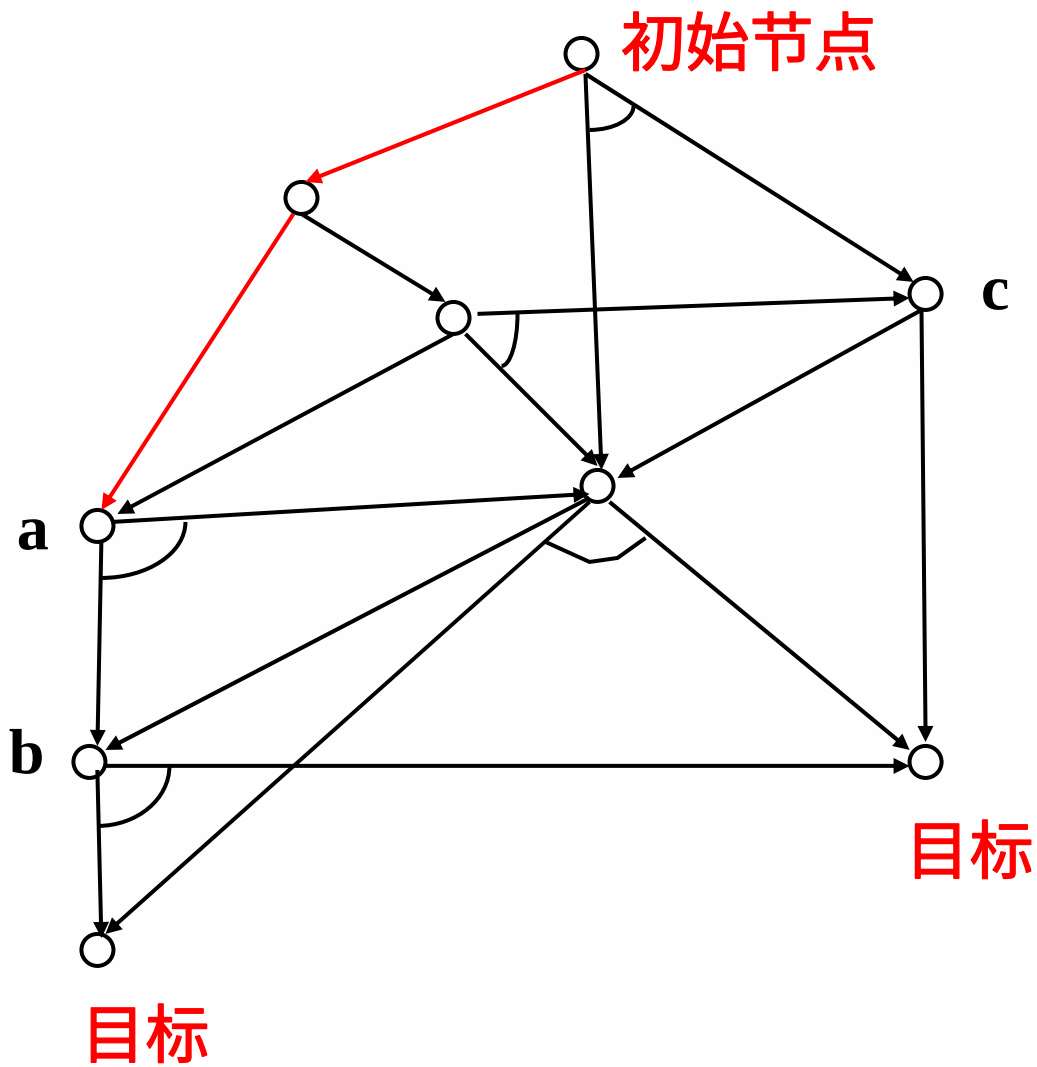
普通图搜索的情况



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

对n的评价实际是对从s到n这条路径的评价

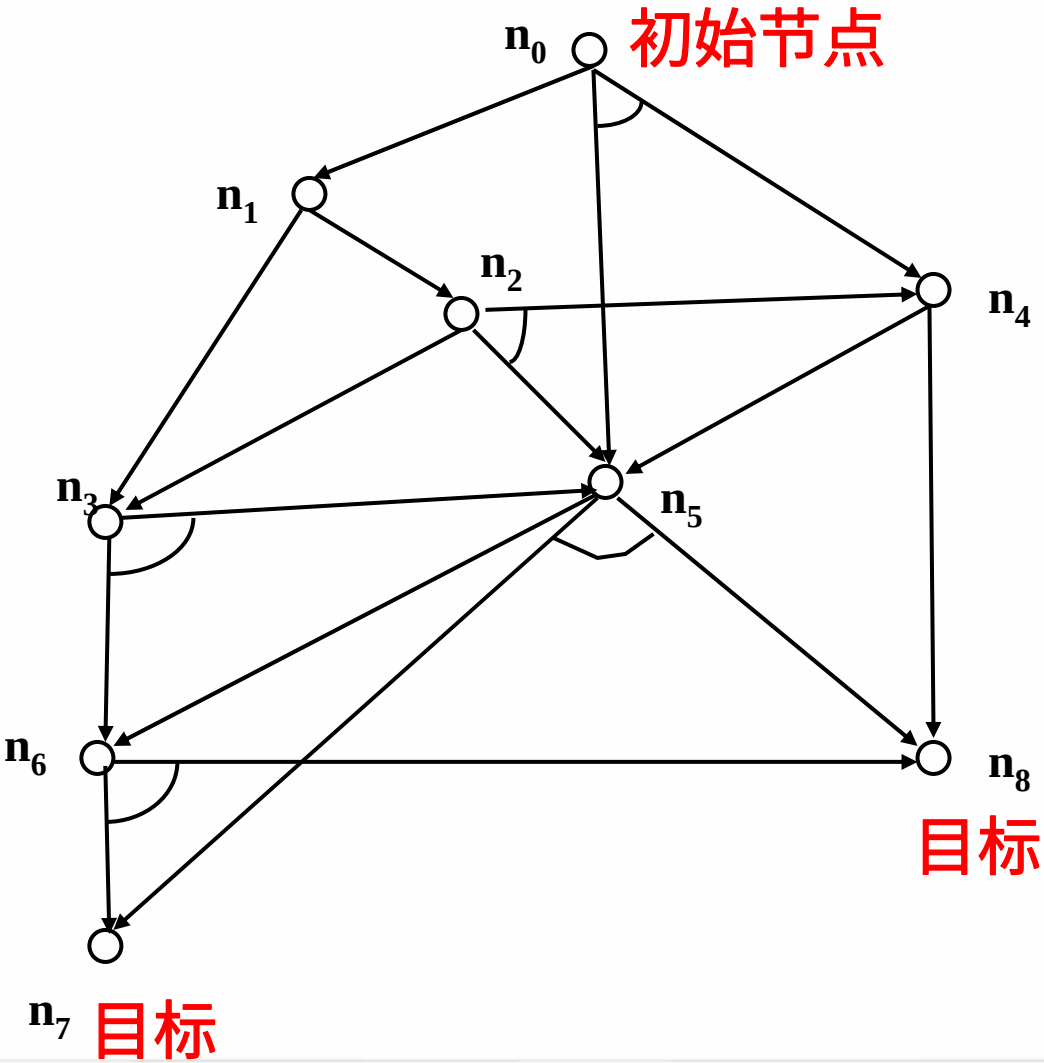
与或图: 对局部图的评价



两个过程

- 图生成过程，即扩展节点
  - 从最优的局部图中选择一个节点扩展
- 计算耗散值的过程
  - 对当前的局部图重新计算耗散值

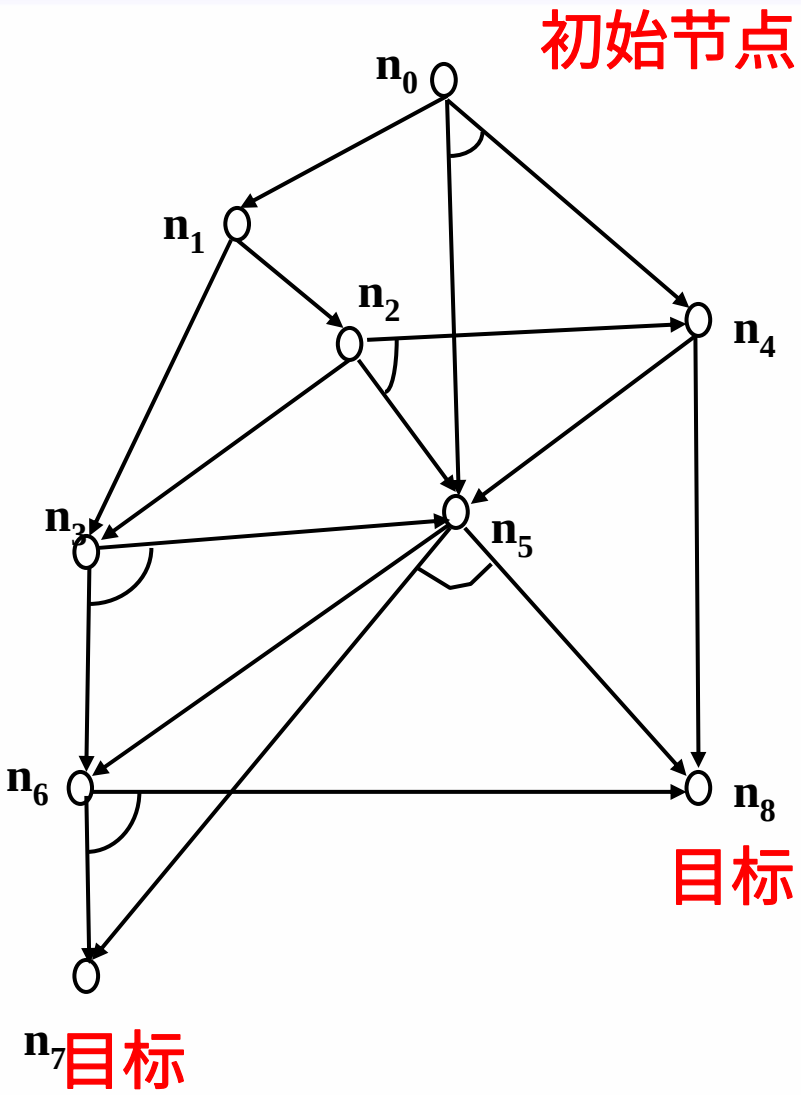
AO*算法举例

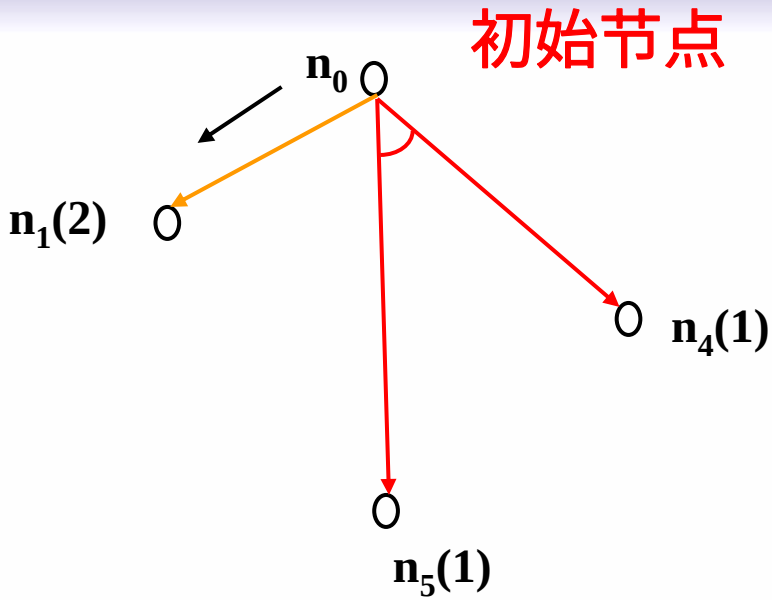


其中：

$h(n_0)=3$
$h(n_1)=2$
$h(n_2)=4$
$h(n_3)=4$
$h(n_4)=1$
$h(n_5)=1$
$h(n_6)=2$
$h(n_7)=0$
$h(n_8)=0$

设：K连接符的耗散值为K

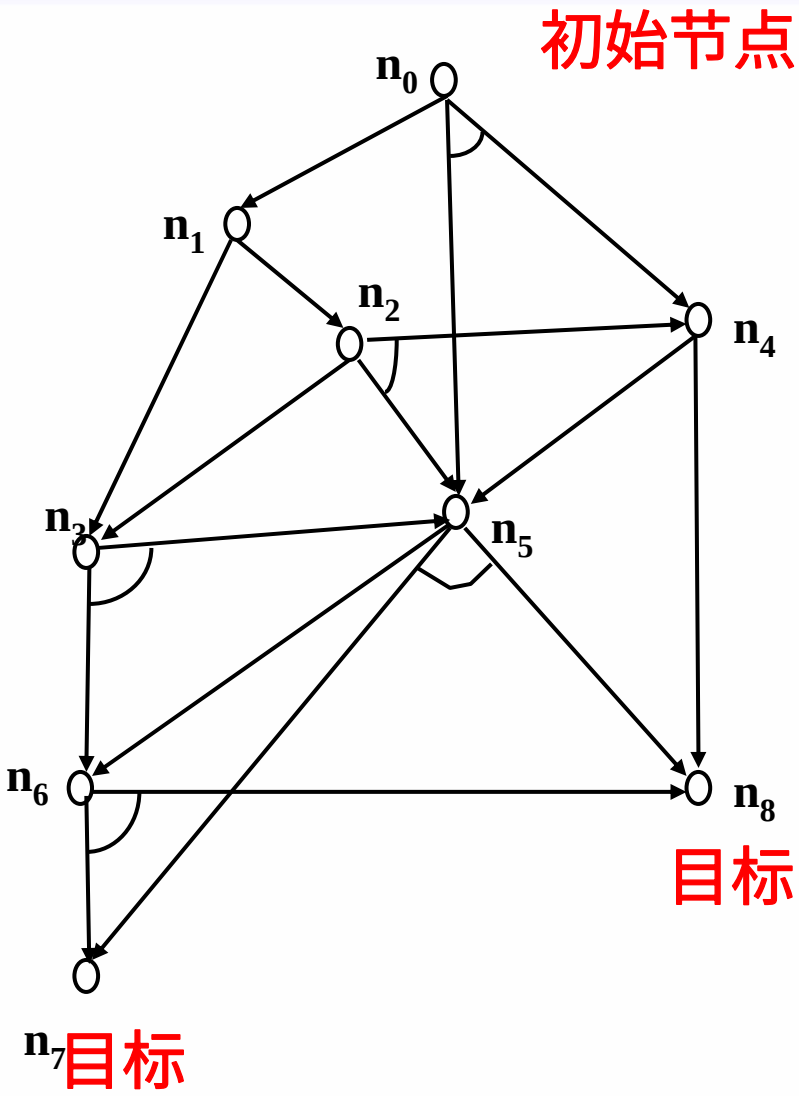


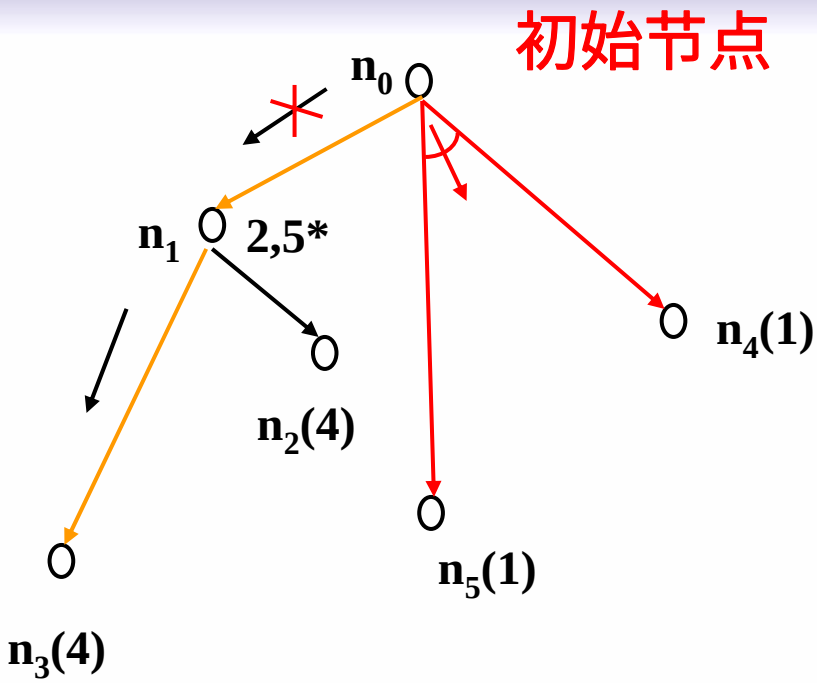


h(n0)=3
h(n1)=2
h(n2)=4
h(n3)=4
h(n4)=1
h(n5)=1
h(n6)=2
h(n7)=0
h(n8)=0

红色：4

黄色：3

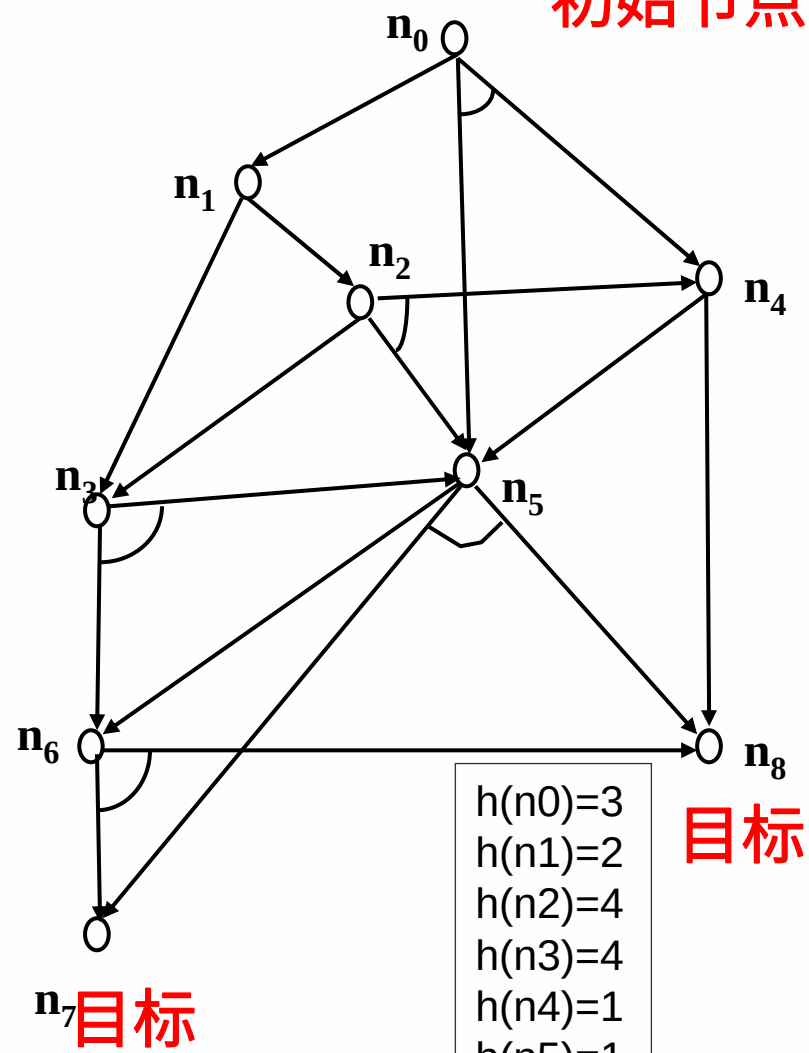


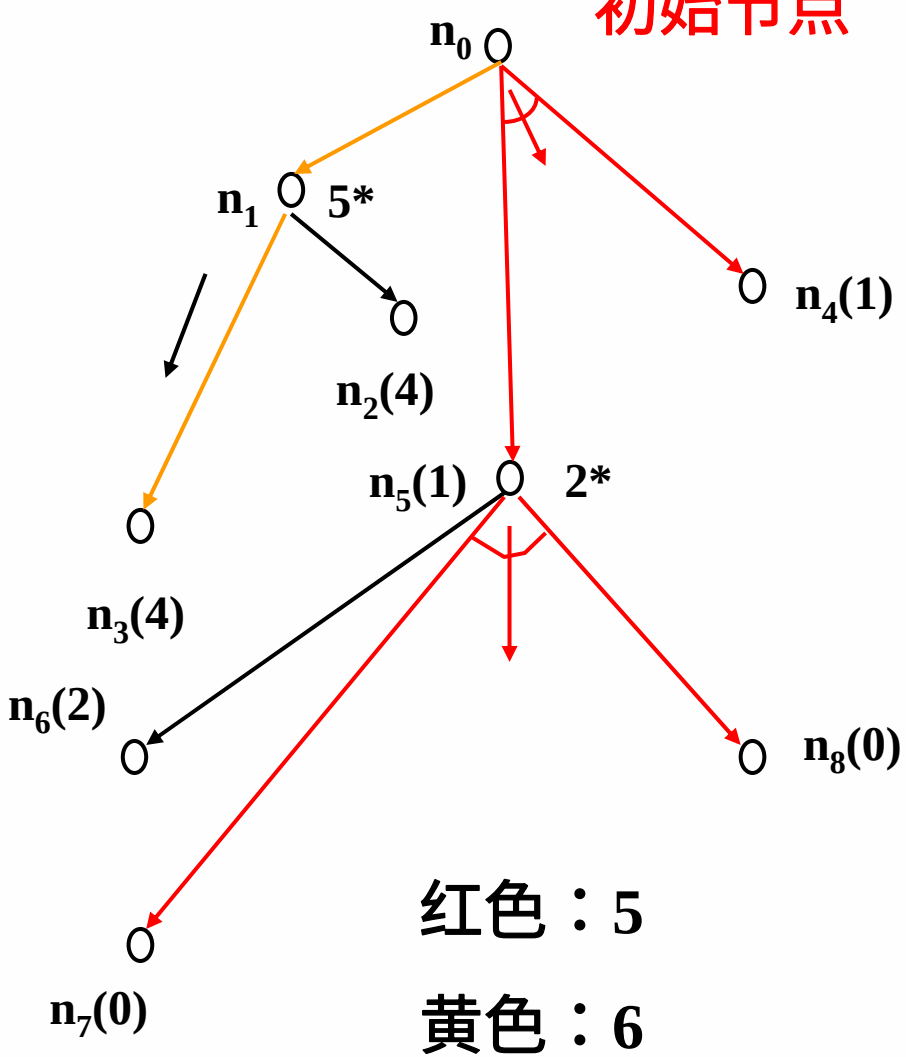


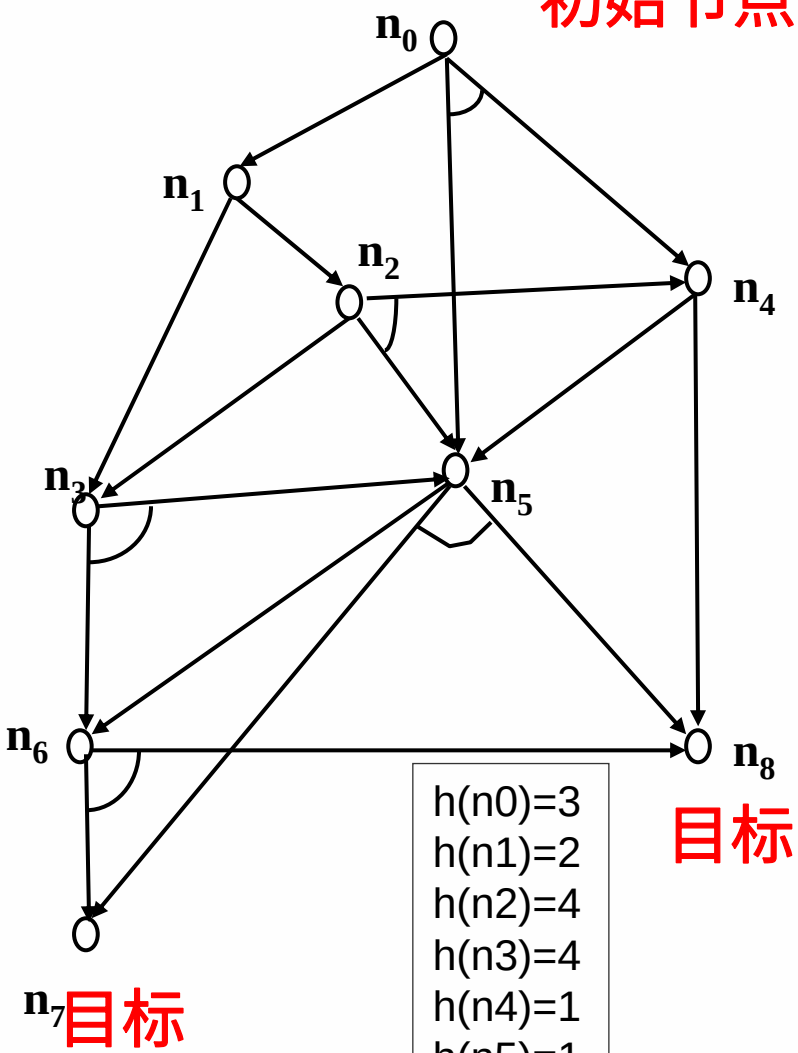
h(n0)=3
h(n1)=2
h(n2)=4
h(n3)=4
h(n4)=1
h(n5)=1
h(n6)=2
h(n7)=0
h(n8)=0

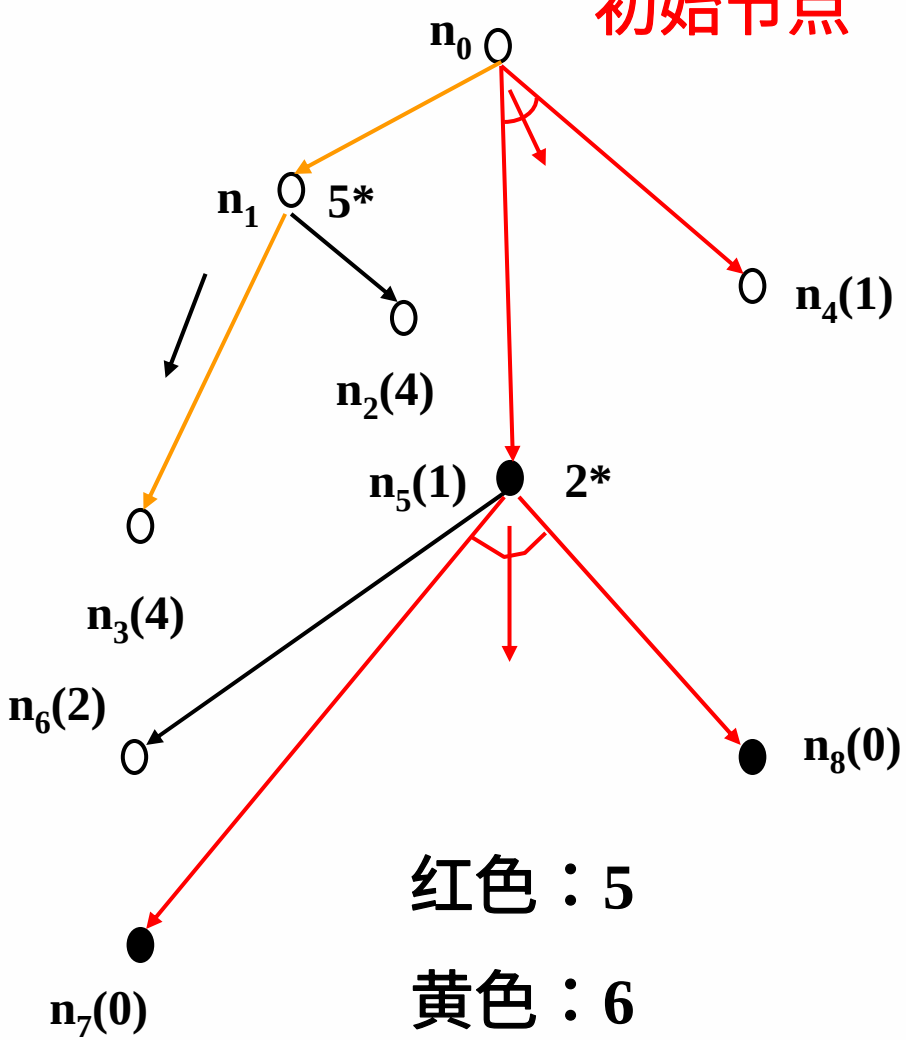
红色：4

黄色：6

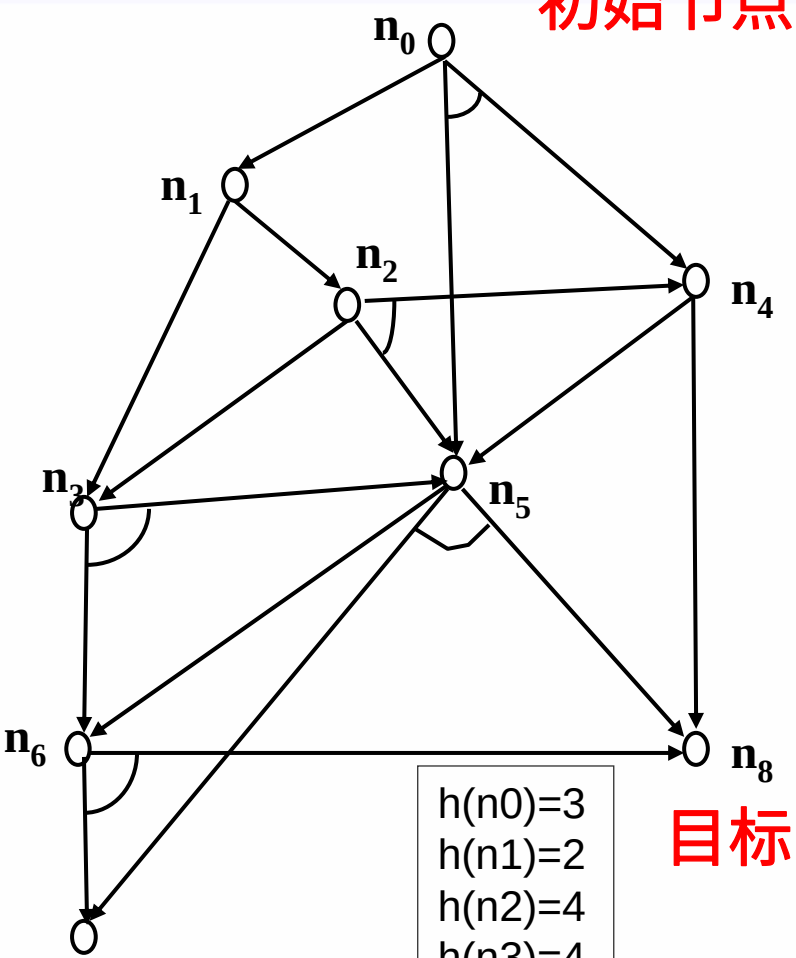

初始节点
n_0
n_1
n_2
n_4
n_3
n_5
n_6
n_8
目标
n_7目标
h(n0)=3
h(n1)=2
h(n2)=4
h(n3)=4
h(n4)=1
h(n5)=1
h(n6)=2
h(n7)=0
h(n8)=0


初始节点
n_0
n_1 5*
$n_4(1)$
$n_2(4)$
$n_5(1)$ 2*
$n_3(4)$
$n_6(2)$
$n_8(0)$
$n_7(0)$
红色：5
黄色：6


初始节点
n_0
n_1
n_2
n_4
n_3
n_5
n_6
n_8
目标
n_7目标
h(n0)=3
h(n1)=2
h(n2)=4
h(n3)=4
h(n4)=1
h(n5)=1
h(n6)=2
h(n7)=0
h(n8)=0


初始节点
n_0
n_1 5*
$n_4(1)$
$n_2(4)$
$n_5(1)$ 2*
$n_3(4)$
$n_6(2)$
$n_8(0)$
$n_7(0)$
红色：5
黄色：6

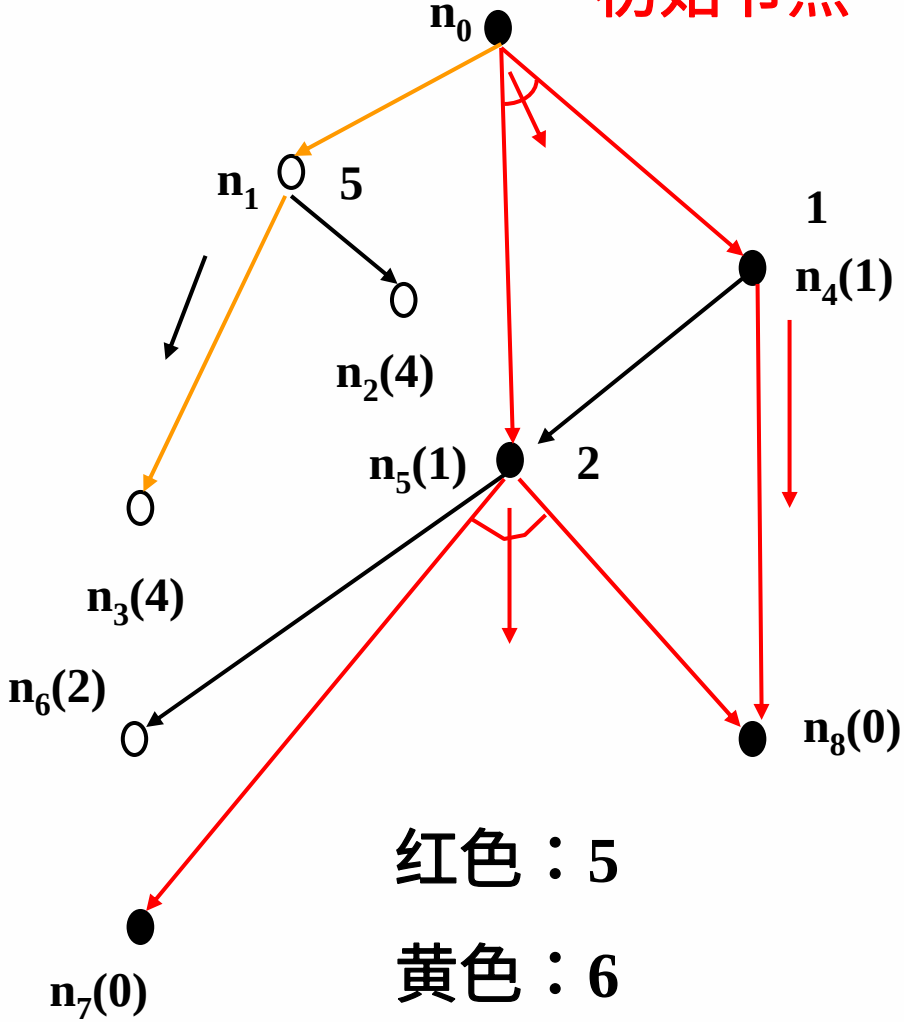
初始节点



目标

h(n0)=3
h(n1)=2
h(n2)=4
h(n3)=4
h(n4)=1
h(n5)=1
h(n6)=2
h(n7)=0
h(n8)=0

初始节点



红色：5

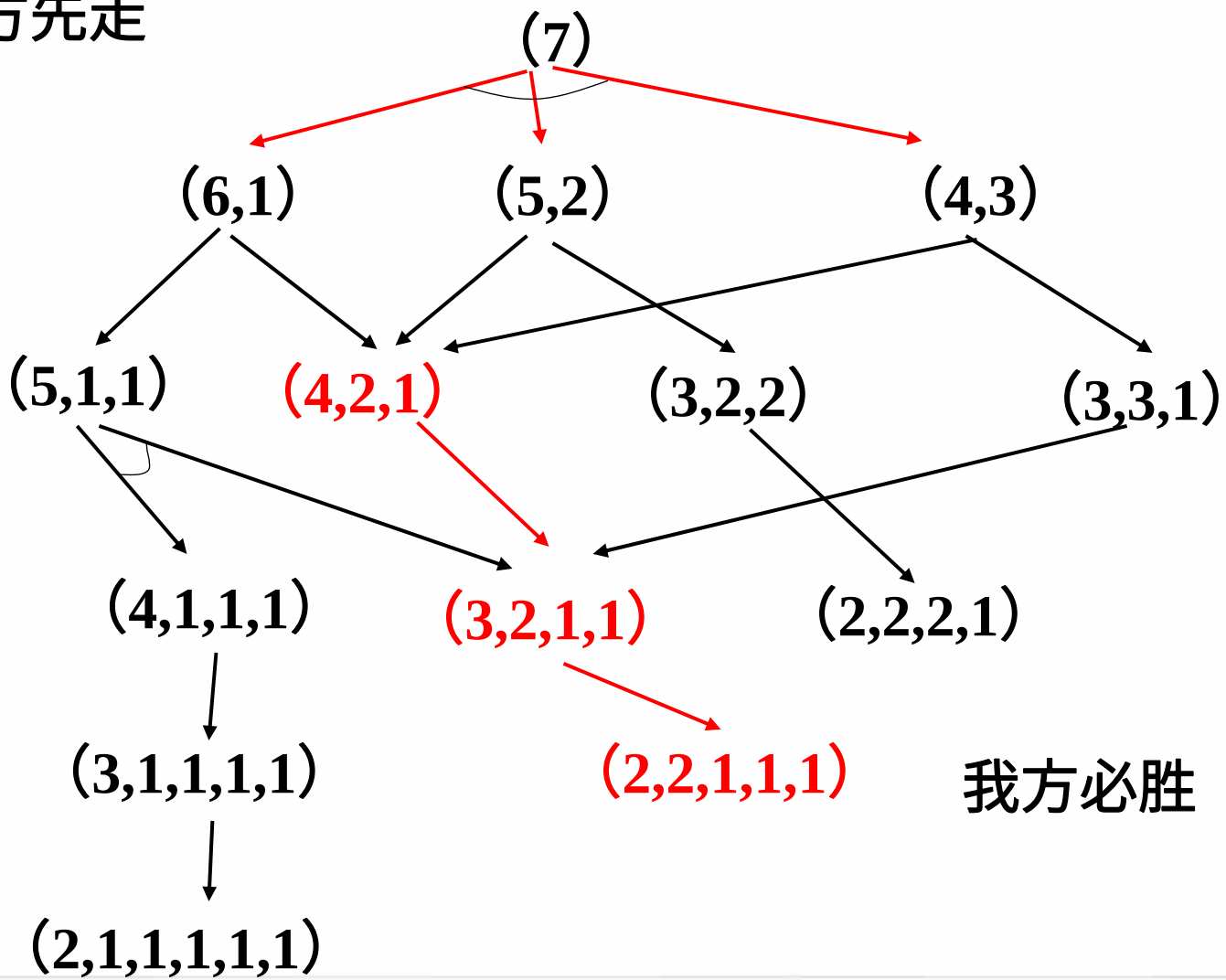
黄色：6

2.3 博弈树搜索

- 博弈过程：
  - 二人零和：博弈结果只有胜、负、平三种情况
  - 全信息：当前及过往局势全开放
  - 非偶然：双方都理智决定行为（选择最有利于己方的策略）
- 博弈树构成
  - 初始格局记为S0
  - “AND”和“OR”逐层交替出现，
  - 己方扩展的为“OR”(MAX)，
  - 对方为“AND”(MIN)
  - 使己方获胜的终局为可解节点，对方获胜的为不可解节点
- 极大极小分析法
  - 端节点用估价函数估值，反推上层节点估值，直至根节点
  - 估值大的方案较好（按获利）
  - 博弈树扩展深度越大越精确，但计算量庞大，通常一定深度

分钱币问题

对方先走

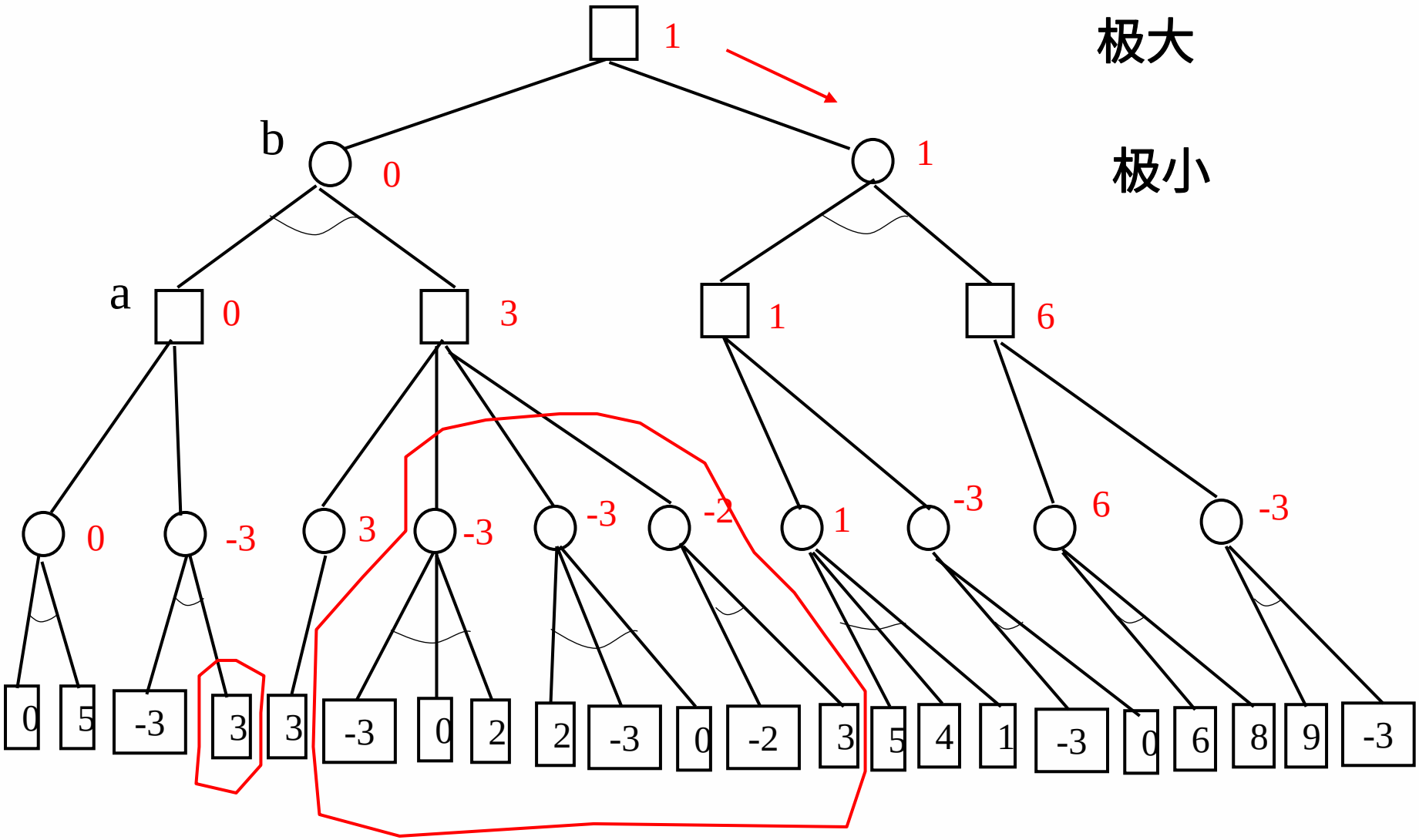


我方必胜

中国象棋

- 一盘棋平均走50步，总状态数约为10的161次方。
- 假设1毫微秒走一步，约需10的145次方年。
- 结论：不可能穷举。

1，极小极大过程



例：一字棋，井字游戏(Tic-Tac-Toe;http://boulter.com/ttt/)

A方：a棋；B方：b棋；扩展深度：2层

估价函数e(p)定义：

$$e(p)=\begin{cases} +\infty, & p\text{是A方胜局} \\ -\infty, & p\text{是B方胜局} \\ e(+p)-e(-p), & p\text{是未定局} \\ 0, & p\text{是和局} \end{cases}$$

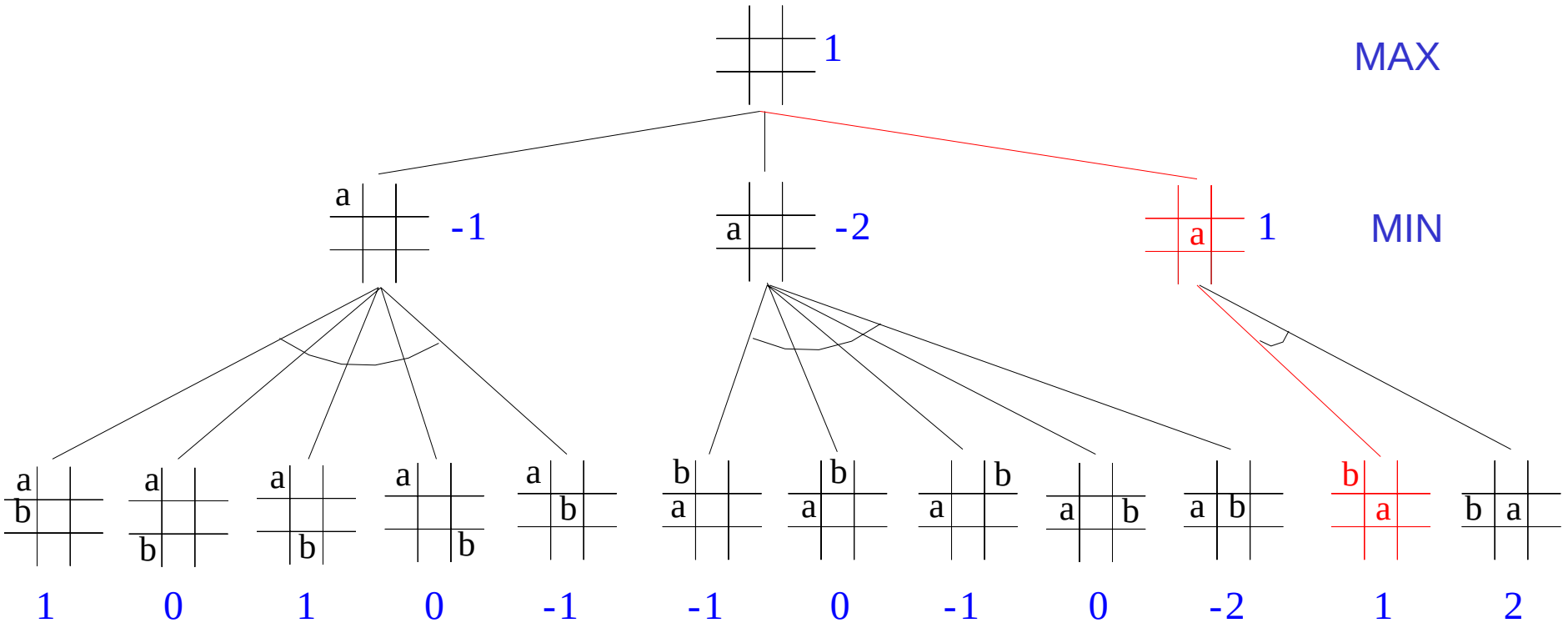
	a	
	b	

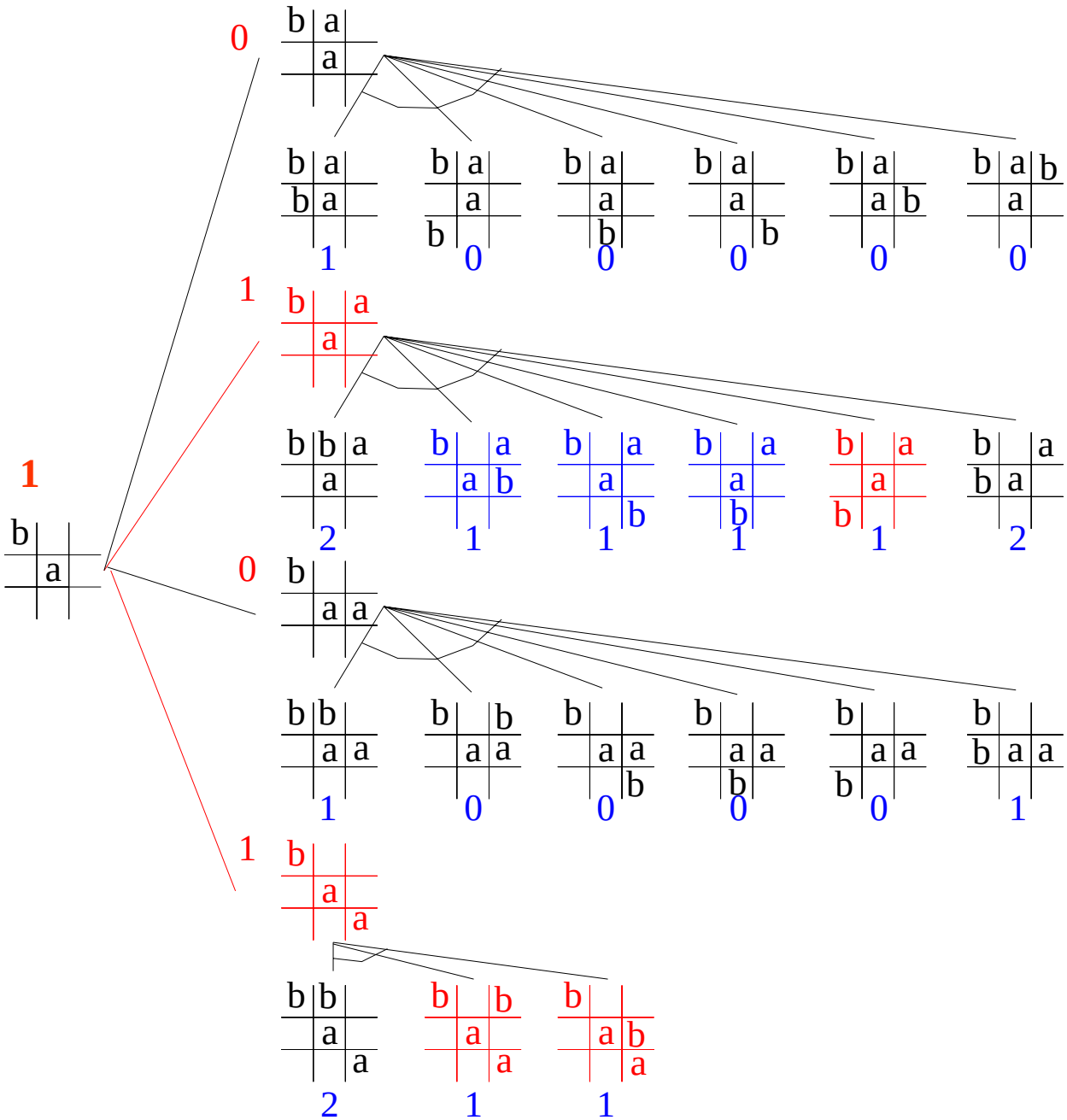
e(+p)：可使a成为三子一线的数目
e(-p)：可使b成为三子一线的数目

a	b	a
b	a	a
b	a	b

5000 states、~900,000 nodes in Tic-Tac-Toe

我方先手

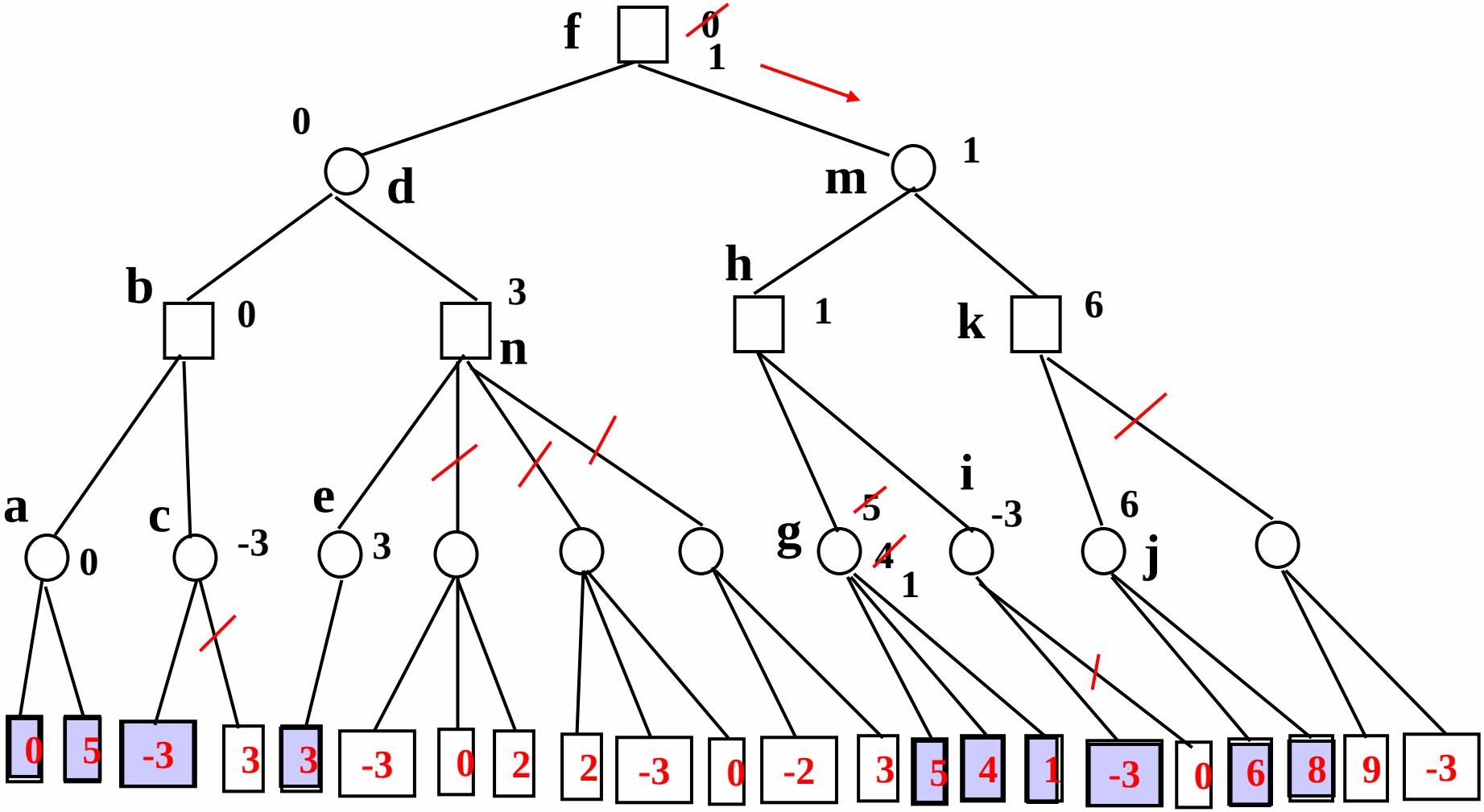

1
MAX
a
-1
a
-2
a
1
MIN
a b
a b
a b
a b
a b
b a
b a
b a
b a
a b
a b
b a
b a
1
0
1
0
-1
-1
0
-1
0
-2
1
2


1
0
1
0
1
1
2
1
1
1
1
2
1
0
0
0
0
1
2
1
1

α-β剪枝

- 边生成估值边计算倒推值，从而剪去某些分支的技术
- 极大节点的下界为α。
- 极小节点的上界为β。
- 顺序：剪枝顺序自左至右或自右至左
- 剪枝的条件：
  - 后辈节点的β值≤祖先节点的α值时， α剪枝
  - 后辈节点的α 值≥祖先节点的β值时， β剪枝
- 简记为：
  - 极小≤极大，剪枝
  - 极大≥极小，剪枝

α-β剪枝（续）


f
0
1
0
d
1
m
b
0
3
n
h
1
k
6
a
0
c
-3
e
3
i
-3
g
5
4
1
6
j
0
5
-3
3
3
-3
0
2
2
-3
0
-2
3
5
4
1
-3
0
6
8
9
-3

Properties of minimax

- Complete? Yes (if tree is finite)
- Optimal? Yes (against an optimal opponent)
- Time complexity? $O(b^m)$
- α-β? $O(b^{m/2})$
- Space complexity? O(bm) (depth-first exploration)

- For chess, b ≈ 35, m ≈100 for "reasonable" games
  → exact solution completely infeasible

Deterministic games in practice

- **Checkers: Chinook ended 40-year-reign of human world champion Marion Tinsley in 1994. Used a precomputed endgame database defining perfect play for all positions involving 8 or fewer pieces on the board, a total of 444 billion positions.**

  »

- **Chess(10^{30}): Deep Blue defeated human world champion Garry Kasparov in a six-game match in 1997. Deep Blue searches 200 million positions per second, uses very sophisticated evaluation, and undisclosed methods for extending some lines of search up to 40 ply.**

- **Othello: human champions refuse to compete against computers, who are too good.**

- **Go: human champions refuse to compete against computers, who are too bad. In go, $b > 300$, so most programs use pattern knowledge bases to suggest plausible moves.**

Annual AAAI GGP Competition

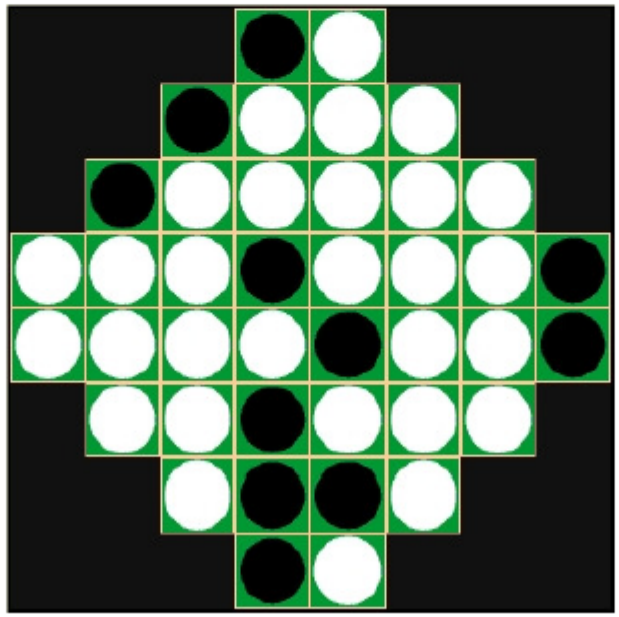
Annual AAAI Competition
- Established in 2005
- Held at AAAI conference
- Administered by Stanford
- (Stanford folks not eligible to participate)

Reward
- Renown
- Satisfaction at a job well done
- Oh … and … $10,000

Nothello

Othello variant

- more corner squares
- **opposite goal:** finish with fewer markers



AAAI-05 Winner Jim Clune

AAAI-06 Winners

Competition at AAAI 2007

Qualification Round via the web
- Weekly tournaments in June (one day each)

Semi-Final Round at AAAI
- Tournament style (one full day)

Final Round at AAAI
- Top two competitors head to head
- Single match of a single game
- Winner take all (glory + $10,000 + chance to …)

Summary

- **Games are fun to work on!**
- **They illustrate several important points about AI**
- **perfection is unattainable → must approximate**
- **good idea to think about what to think about**

问题1：12枚硬币，其中有一枚假币，（凡轻于或重于真币者即为假币），设计一个算法，使用无砝码天平至多称3次，找出假币并指出它轻于还是重于真币。

问题2：任给k枚硬币，其中有一枚假币，问至少称几次可以找出假币。

现有标号为1，2，3，4，5，6，7，8，9的九张纸牌，甲、乙两人轮流取一张牌，已取走的牌不能重新放回，目标是手中任意三张牌的点数加起来正好是15点，谁先拿到，谁就赢了。