by tianfangjh
sch

Google Camp

//搭建输入输出框架

//完善估值函数

搜索算法

性能优化

Google Camp

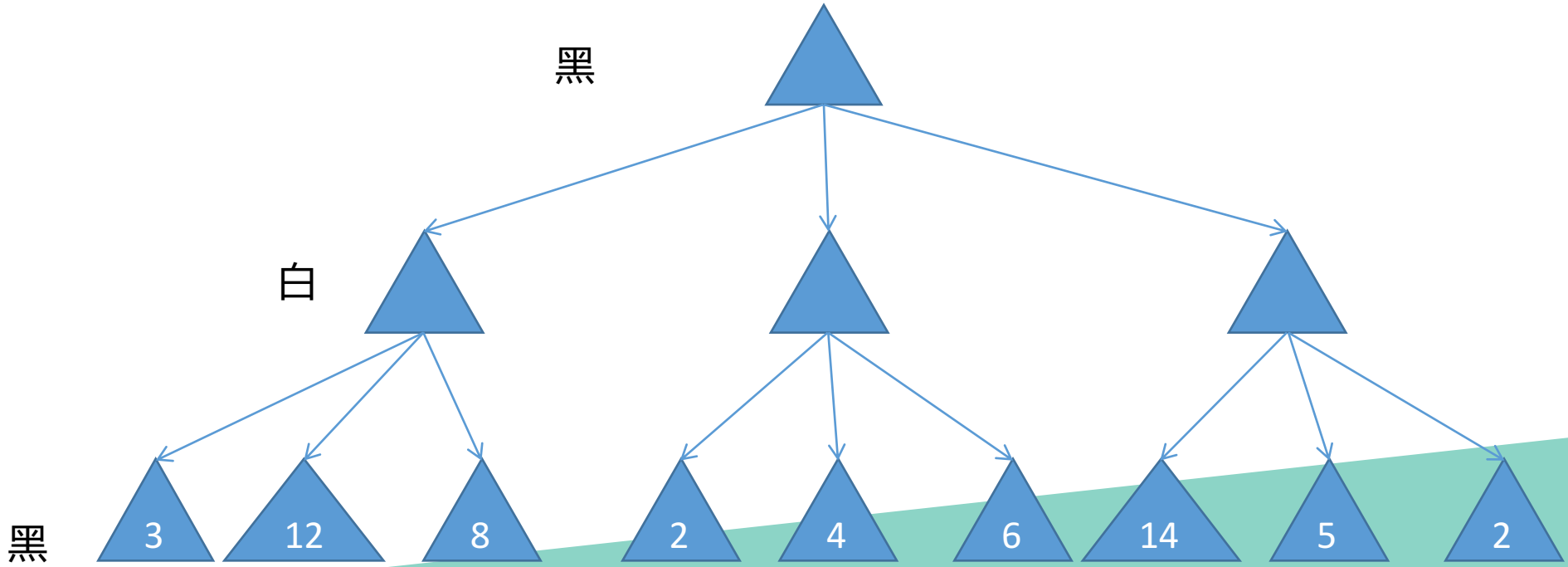
人是怎么下棋的

1.0 思考下一步的最佳位置（局部）

贪婪算法

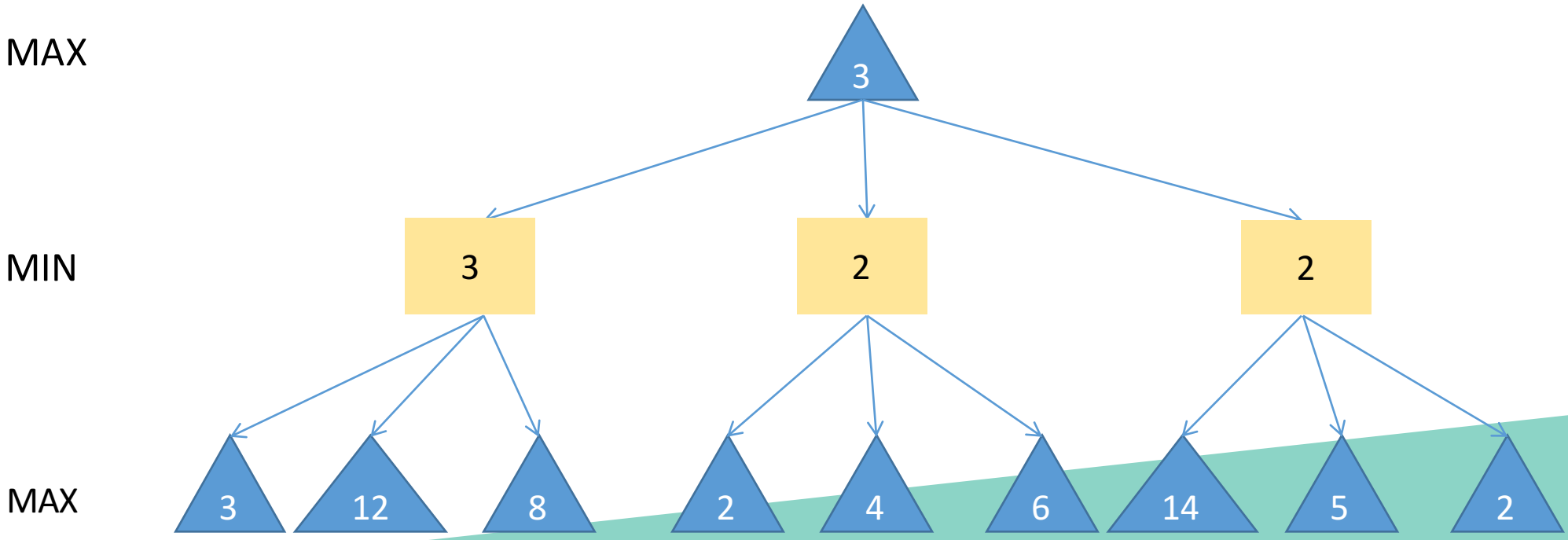
2.0 多考虑几步？（相对的全局）

深度优先搜索



Google Camp

Minimax

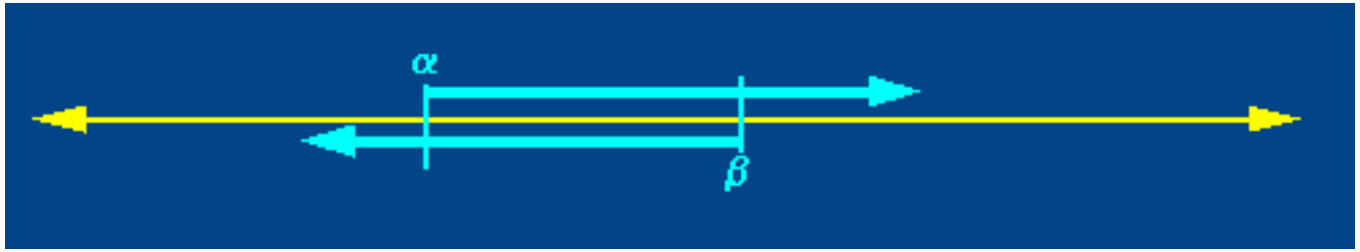


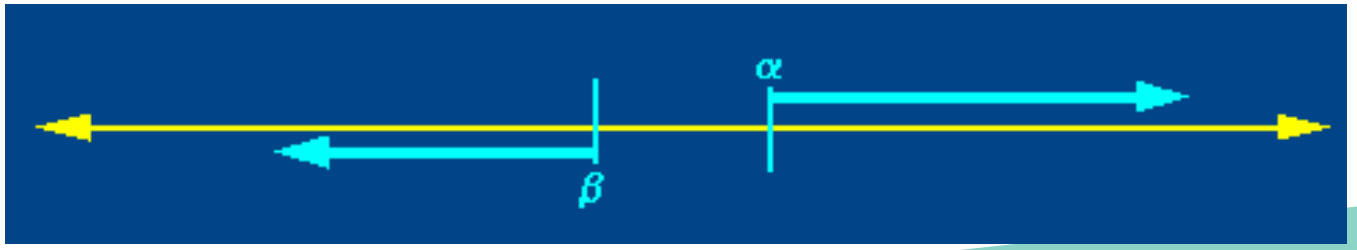
Minimax with Alpha Beta Prunning

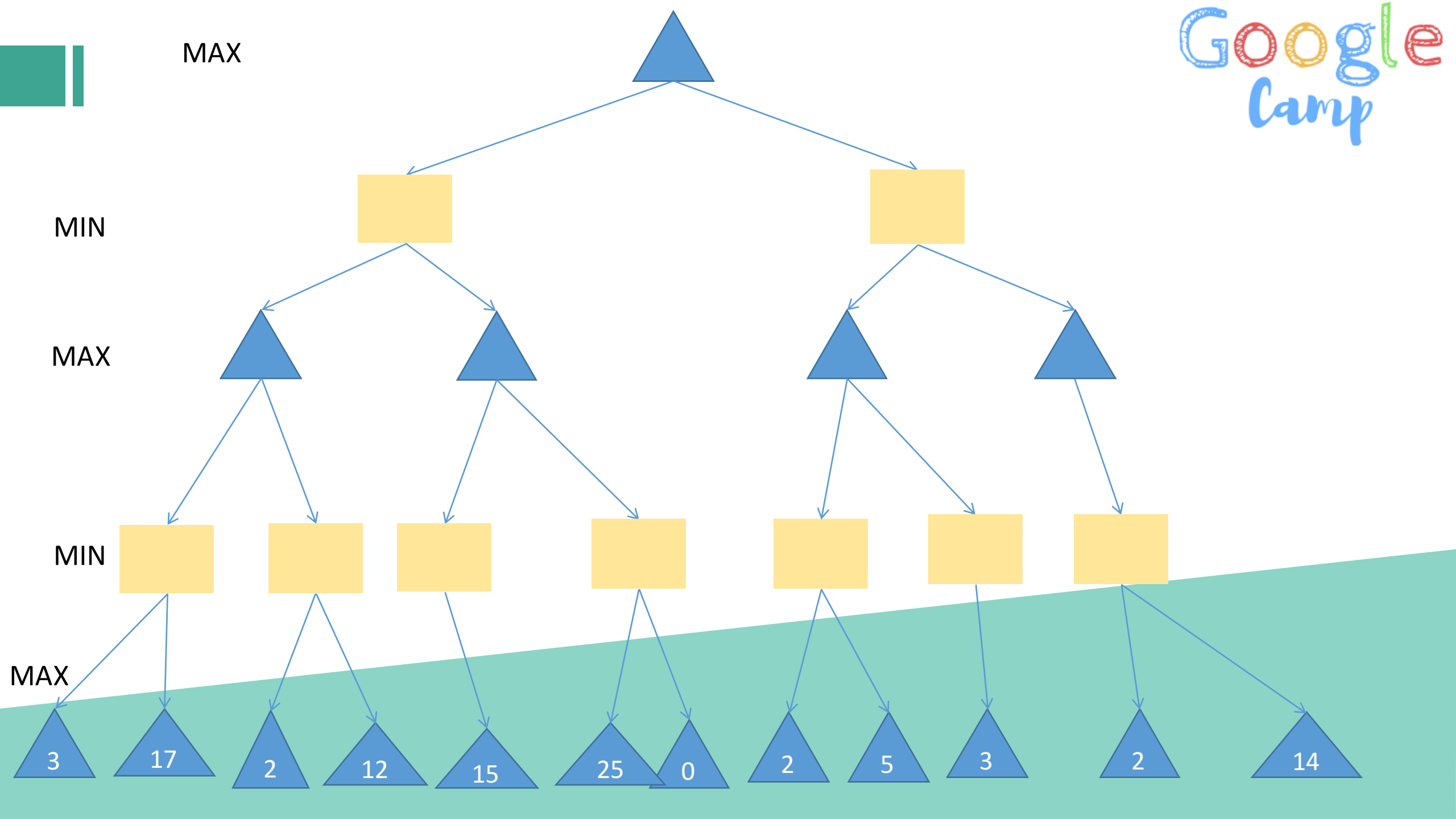
α :可行解的最大的下界

β :可行解的最小的上界

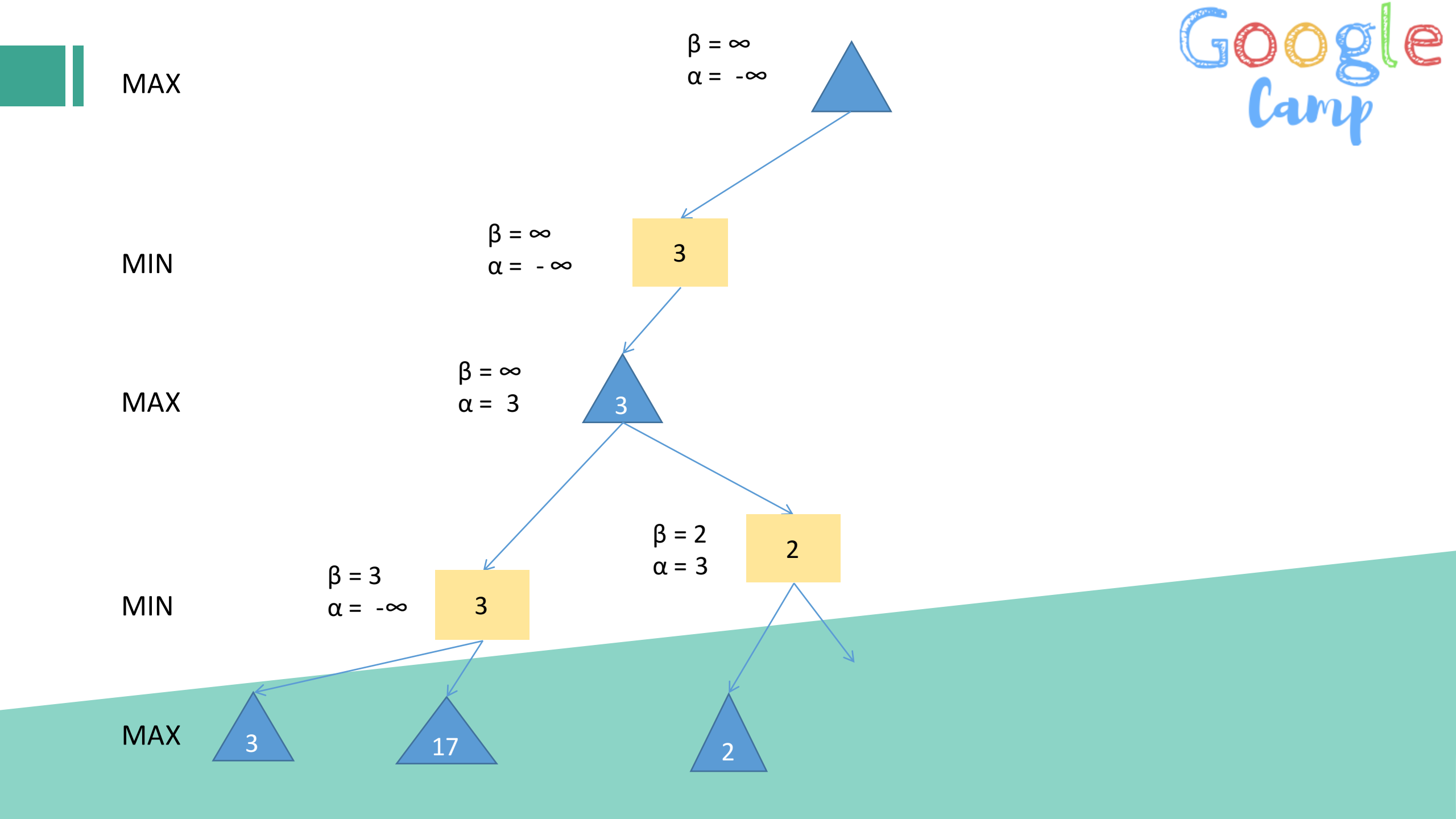

α
β

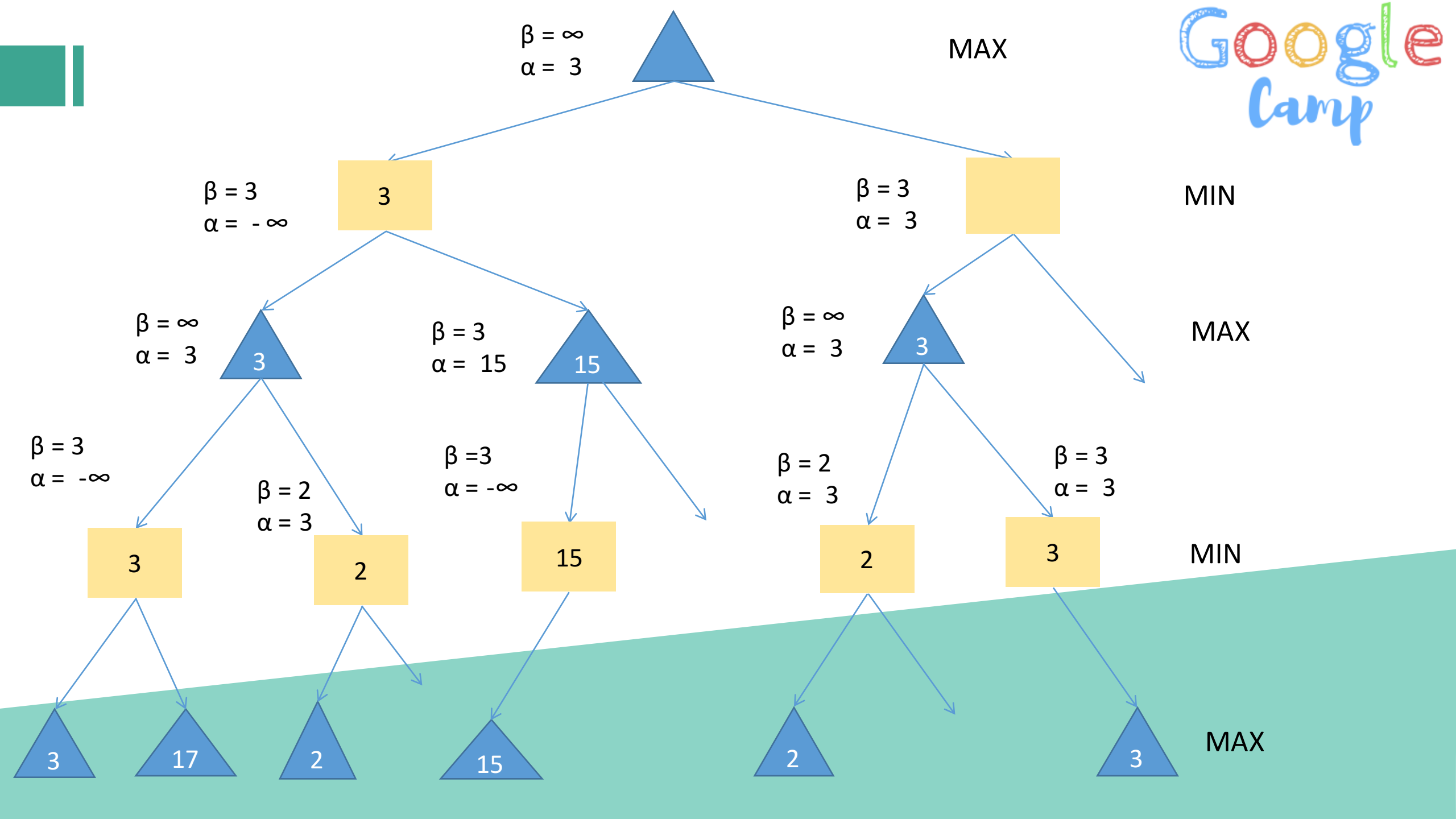

α
β


α
β


MAX
MIN
MAX
MIN
MAX
3
17
2
12
15
25
0
2
5
3
2
14

剪枝


MAX
β = ∞
α = -∞
MIN
β = ∞
α = - ∞
3
MAX
β = ∞
α = 3
3
MIN
β = 3
α = -∞
3
β = 2
α = 3
2
MAX
3
17
2


β = ∞
α = 3
MAX
3
β = 3
α = - ∞
β = 3
α = 3
MIN
β = ∞
α = 3
3
β = 3
α = 15
15
β = ∞
α = 3
3
MAX
β = 3
α = -∞
β = 2
α = 3
β =3
α = -∞
β = 2
α = 3
β = 3
α = 3
3
2
15
2
3
MIN
3
17
2
15
2
3
MAX

Google Camp

α- β剪枝是否已经优化到底了？

重复搜索

[3,4],[8,7],[7,6],[7,9]
[7,6],[7,9],[3,4],[8,7]

局面->分值

Address = Hash(key)

Google Camp

棋盘特征

- 棋盘的位置
- 位置上棋子的状态

Zobrist算法

异或运算

- 结合律：RI XOR (RJ XOR RK) = (RI XOR RJ) XOR RK
- 交换律：RI XOR RJ = RJ XOR RI
- RI XOR RI = 0
- RI XOR 0 = RI

Zobrist键值

- 两个Zobrist[size][size]的数组
  数组的每一个都填上一个随机数，至少32位
  键值等于棋盘上所有棋子对应的随机数的异或运算

- 移动棋子
  K XOR R(自,x0, y0)
  K XOR R(自,x, y)
- 吃子
  K XOR R(反,x,y)
  K XOR R(自,x,y)
- 生成一个随机数表示走棋方

- K XOR R(side)

哈希表

Google Camp

- **Address = Hash(key)**

  ZobristKey() % tablesize()

  tablesize最好为2^n;

  int tablesizemask = tablesize() – 1;

  zobristkey() % tablesize() =
  zobristkey() & tablesizemask;由来

- **存储内容**

  搜索深度、局面估值、Zobrist键值

- **冲突**

  线性探测再散列

  H_0=H(key)
  H_{j+1}=(H_j+1)%m; j = 1, 2, …

- **取用**

  搜索一个节点时，尝试从置换表中取出结果，如果存在且深度符合，键值相同，就可以使用

Attention

- 解决冲突

  线性探测再散列

  始终覆盖

  深度优先策略

- 合理设置哈希表的长度

Google Camp

goto 吹水时间;//（逃

一些可能有、用的小tips

0.用short char 代替 int

0.0 嗯，该炸的还是会炸

1.我为什么要malloc/calloc呀

1.0 好像和保存现场和回溯有关…

1.1 不要忘了free…

2.我为什么要用 memcpy/memset/memmove

2.0 可能会快点

2.1 容易理解->所谓抽象（付大佬会讲

~~2.2 看起来比较高端？~~

Google Camp

怎样稍微优雅一点地申请内存？

0.锯齿形数组(Ragged Array)

- 0.0 啥是锯齿形数组
- 0.1 怎么弄呀
- 0.2 可以干啥 ->模拟棋盘

1.一维还是二维？

- 1.1 ？？？棋盘不是二维吗
- 1.2 二维数组的内存组织方式

Google Camp

- 所以你应该可以用dynamic的一维数组和memcpy去完成搜索中内存相关的那部分

- 生活不是只有大项目，还有上机考

- 好的，告辞0.0