

きふわらべ

アピール文書

2022年01月28日 高橋智史

GO

囲碁を知っているふりをする きふわらべ で出場



開発者 (きふわらべのお父ん)

高橋 智史 (左画像に著作権上、問題ありません)

「UCBI を使った モンテカルロ木探索 で出場するぜ。
2月中に がんばれば独自性として アキ三角サカレガタチ理論 と
マジックナンバー $2\pi e$ (にいパイイー) の2つを使って
実装するかも知れないし、間に合わないかも知れないぜ」



コンピュータ囲碁エンジン

きふわらべ (左画像に著作権上、問題ありません)

「講習会のサンプルプログラム ベースだよな。
間に合うか、間に合わないかを 教えるだけ」



ひよ子 (左画像に著作権上、問題ありません)

「例年の様子だと、間に合わないのよねえ」

このPR文書で使われているフォント (利用ライセンス上、問題ありません)

フリーフォント「たぬき油性マジック」 作者：たぬき侍

<https://tanukifont.com/tanuki-permanent-marker/>

KOSHUKAI

2015年7月~9月 これで書ける! 「コンピュータ囲碁講習会」

これで書ける! 「コンピュータ囲碁講習会」 (利用ライセンスのようなものは明記なし)

http://entcog.c.occo.jp/entcog/cg_koushu.html



「👉 2016年3月に AlphaGo対李世ドル があって そこから
ディープラーニングが流行り始めたが、
講習会は その1年前だから ニューラルネットも使ってないぜ」



「お父ん、ディープラーニング分からないから
最強ソフトは プロとは4子の手合い、と言われた時代の
入門者向けの サンプルプログラム まだ いじってる」

ごご

2019年11月 GoGo

GoGo (Twitterで利用許可確認済み)

<https://github.com/bleu48/GoGo>



「👉 2019年11月に しば のおっちゃんが
講習会の C言語 のコードを Go言語 で再実装したんだぜ。
2020年に わたしがそれをいじって壊して
第12回UEC杯に出場したんだぜ」



「今年のコードも それベースよね」

FUNIKI

2022年1月 Kifuwarabe-UEC13 開発中の雰囲気

```
File Edit Selection View Go Run Terminal Help playout.go - kifuwarabe-uec13 (Workspace) - Visual Studio Code
EXPLORER
OPEN EDITORS
x playout.go entities M
KIFUWARABE-UEC13 (WORKSPACE)
kifuwarabe-uec13
coding_obj
config_obj
doc
entities
board_i.go
board.go
get_winner.go
global.go
monte-carlo-tree-search.go
parameter_adjustment.go
playout.go M
put_stone.go
record_item.go
uct.go
input
output
presenter
kifuwarabe-uec13 > entities > playout.go > Playout
7 // Playout - 最後まで石を打ちます。得点を返します
8 // * `getWinner` - 地計算
9 //
10 // # Returns
11 //
12 // 手番が勝ったら 1、引分けなら 0、相手が勝ったら -1
13 func Playout(
14     board IBoard,
15     turnColor int,
16     getWinner func(int) int) int {
17
18     AllPlayouts++
19
20     var color = turnColor
21     var previousZ = 0
22     var boardMax = board.SentinelBoardArea()
23
24     var playoutTrialCount = PlayoutTrialCount
25     for trial := 0; trial < playoutTrialCount; trial++ {
26         var empty = make([]int, boardMax)
27         var emptyNum, r, z int
28
29         var onPoint = func(z int) {
```


```
30
31     if !board.Exists(z) {
32         empty[emptyNum] = z
33         emptyNum++
34     }
35
36     board.IterateWithoutWall(onPoint)
37
```

play white S18
gtp_z=S18
x=18 y=18 z=396 z4=1818

	A	B	C	D	E	F	G	H	J	K	L	M	N	O	P	Q	R	S	T	
1
2
3	.	.	x
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19

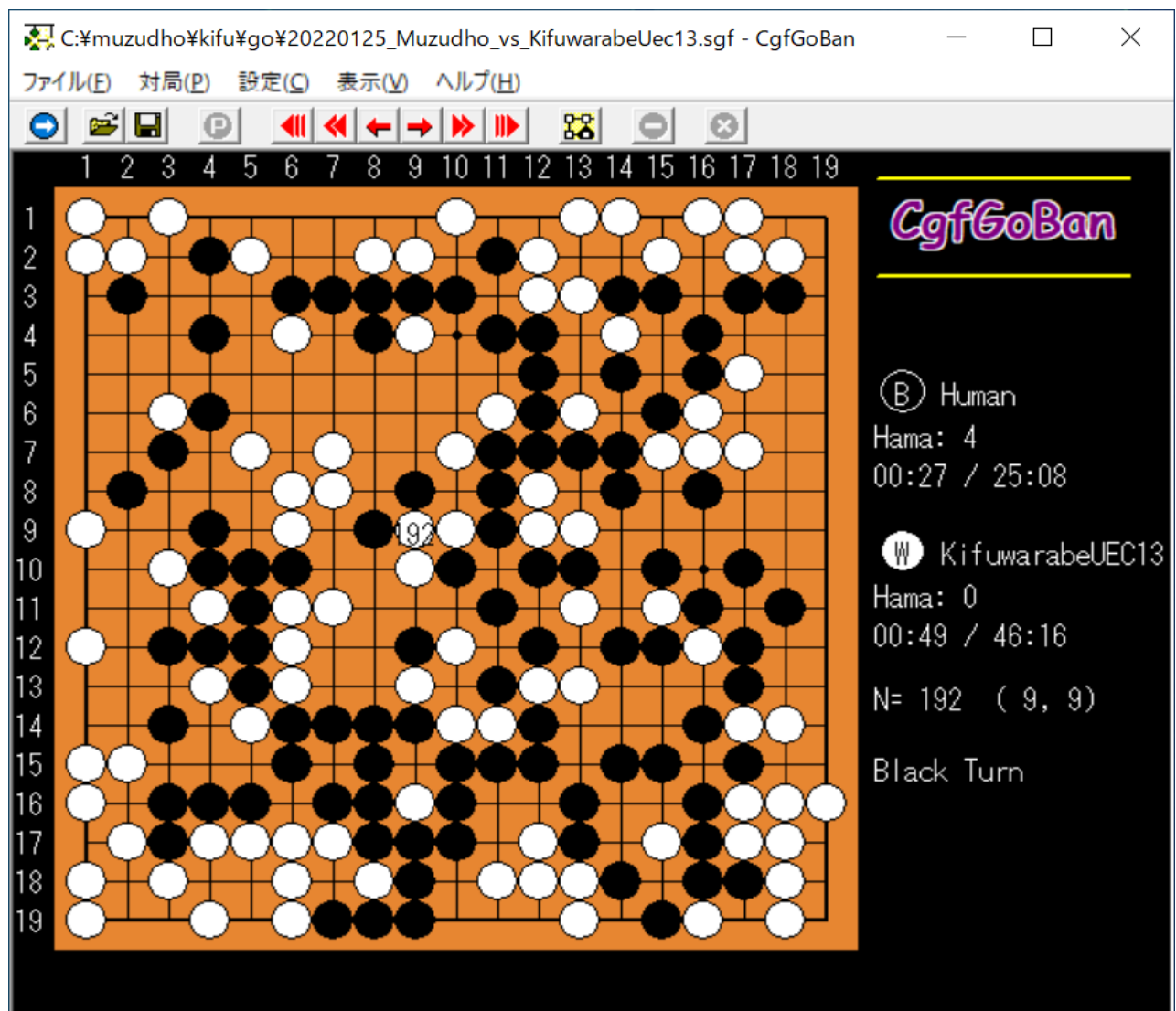
KoZ=0, movesNum=2
=
quit



「 ソースをいじっている様子だぜ。
やってることは 講習会の レッスン9 ぐらいまでだぜ」



「申込期限まで もう時間ないぜ、さっさと出せだぜ！
PR文書も添付しないと 申込を提出できないんだぜ」



「 まだ **アキ三角サレガタチ理論** を入れてないぜ。
わたしが打っても 勝てるぜ」



「早く 独自性を 入れましょう！」



「**アキ三角サレガタチ理論** は わたしの考えた理論じゃないけど
 $2\pi e$ を使うのは わたししか やらないだろ。
今年は 独自性があるかも。 実装すれば」

アキ三角サカレガタチ理論

依田紀基チャンネル アキ三角サカレガタチ理論（序章）
～「筋場理論」改める。なぜこの講義を始めるに至ったか～（利用ライセンス不明）

<https://www.youtube.com/watch?v=FXqfJZVky0A>

依田紀基チャンネル 「アキ三角サカレガタチ理論」本編
～筋の新定義！アキ三角とサカレガタチは表裏一体～（利用ライセンス不明）

<https://www.youtube.com/watch?v=p4G1HnyYVWc>



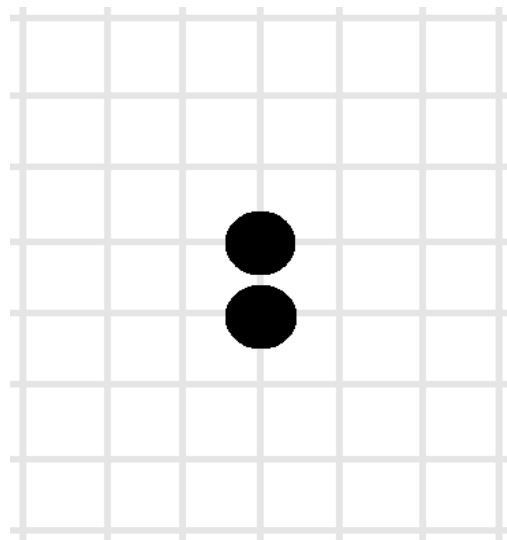
「☞ この理論を 使おうぜ」



「 使用の許可は……」



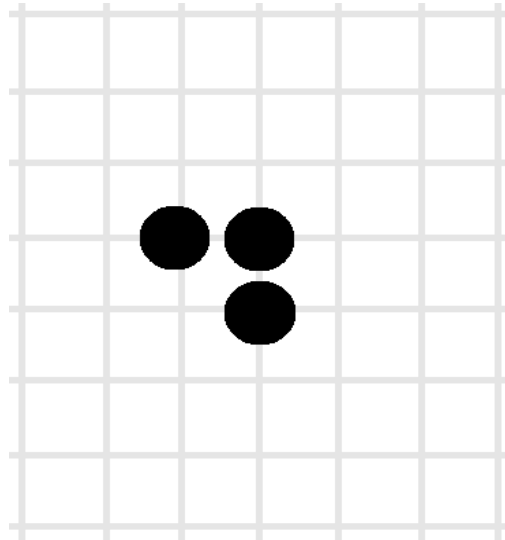
「 許諾は得てないけど、 動画に何にも使用制限が書いてないことは 確認したわよ」



「☞ 黒石が2つ並んでいるときに、
次に絶対に 黒石を置いてはいけない場所があるそうだけ」



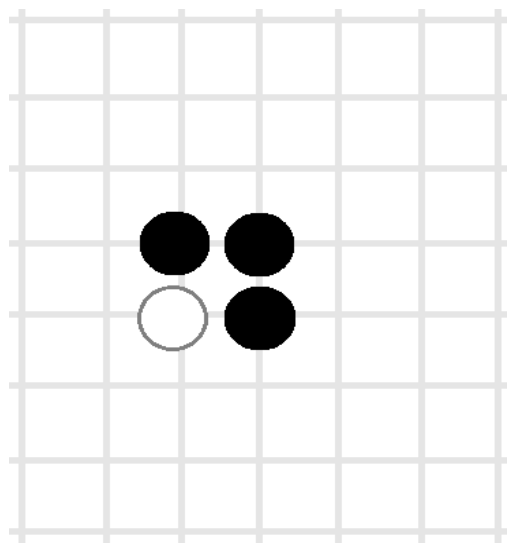
「悪手の理論かだぜ」



「☞ それがこの アキ三角 という形だそうだぜ。
角が1つ欠けてるな」



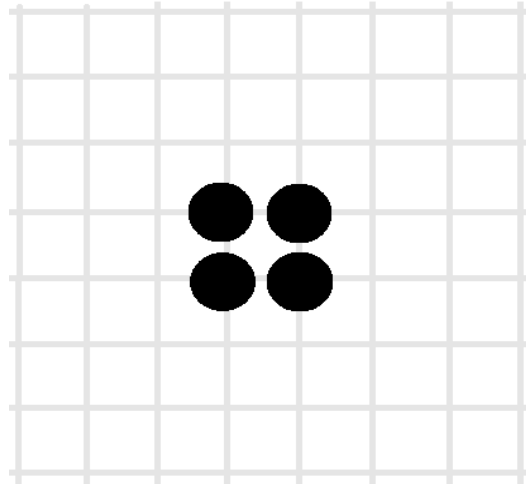
「連珠みたいねえ」



「☞ 角に 相手の石があるときは アキ三角 ではないぜ。
これは黒は マガリ と言って 悪形ではないそうだぜ」



「じゃあ 相手の白は アキ三角 の候補地を1個潰してしまって 相手を助けてしまっているんだな」



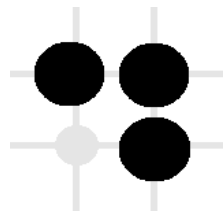
「☞ アキを埋めて 四角にしてやったら どうなの？ この形は 悪形 なの？ 悪形は消えてるの？」



「アキ三角は 石は離して打て、ということ を 言ってるんだぜ。好きで こんなに くっつけて打つことないだろ。」

めんどくさいから

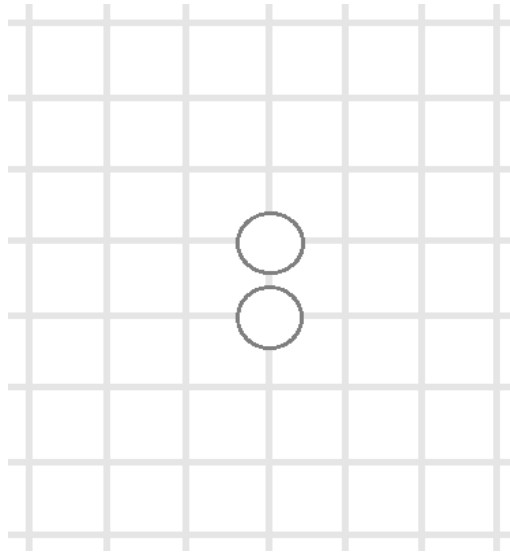
悪形 ということでもいいんじゃないか」



「じゃあ 空点または黒石 を明示するために 小さな灰色の丸 という記号を導入しましょう」



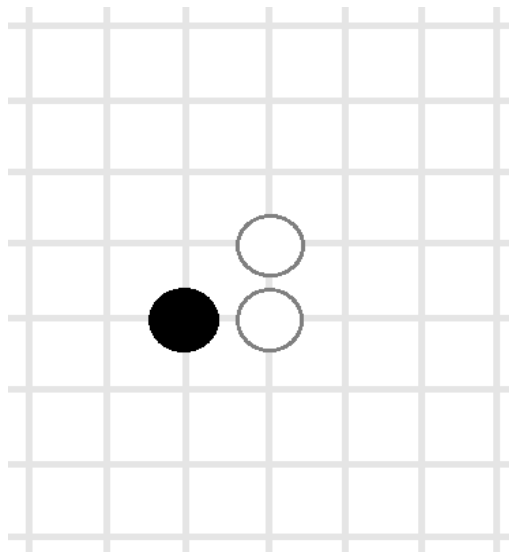
「考えるのは めんどくさがるのに 書くのは細かいな」



「☞ 逆に 相手の白石が2つ隣接してるとしよう。
ここで 黒石が次に置くのは 良くはない場所があるそうだけ」



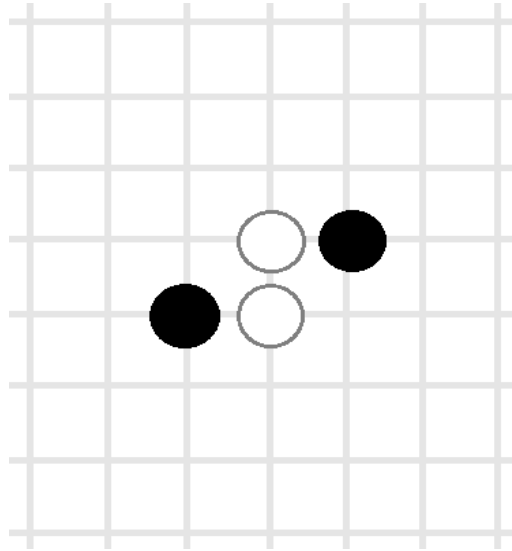
「さっき言った 相手を助ける手よ」



「☞ まあそうだな。そして上図で 黒の次の一手で
絶対 置いてはいけない箇所が 1つ あるそうだけ」



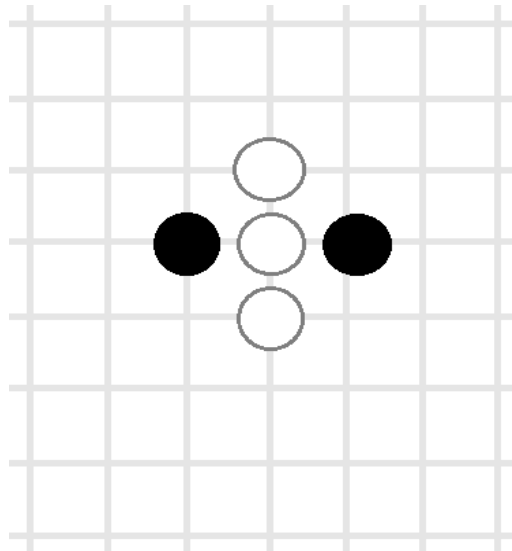
「将棋の一手は 悪手の中からマシな手を1つ選ぶこと ばかりなのに、
囲碁の一手は 絶対置いてはいけない悪手って 1個ぐらいなんだな」



「☞ ケイマが ぶった切られたような形が
目も当てられないような 悪形らしいぜ。
こういうの サカレガタチといって 2種類ある」



「線を伸ばすゲームなのに 切れてますもんね」



「☞ これも サカレガタチ。 サカレガタチは この2つだけ」



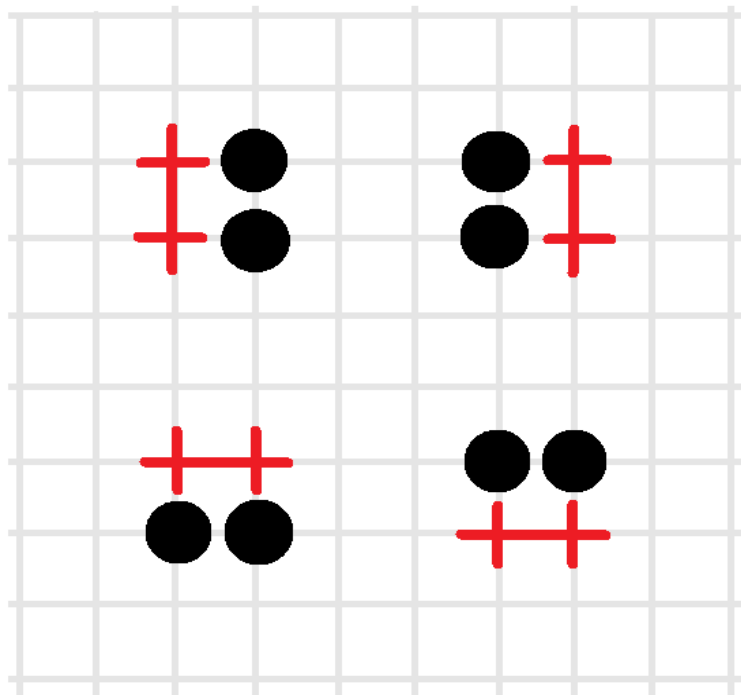
「よし 悪形は暗記したぜ。 どうやって 実装するんだぜ？」



「UCT1 や モンテカルロ木探索と からめられるの？」

NINSHIKI

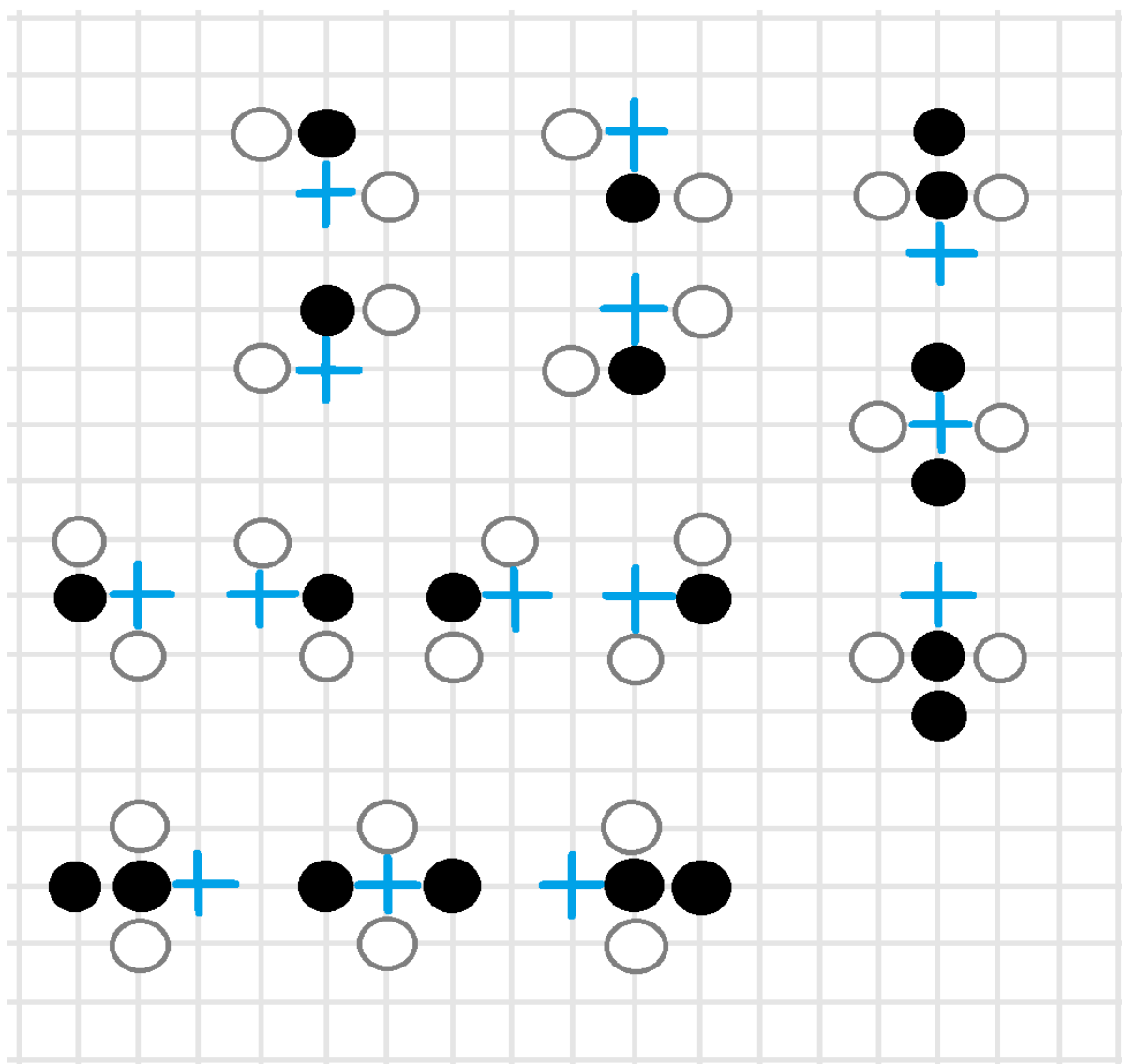
アキ三角サレガタ手を パターン認識 できるの？



「☞ 4つの交点を見て、黒石2つ、空点2つ
隣接している図形の 空点は 黒石から見て悪手だろ」



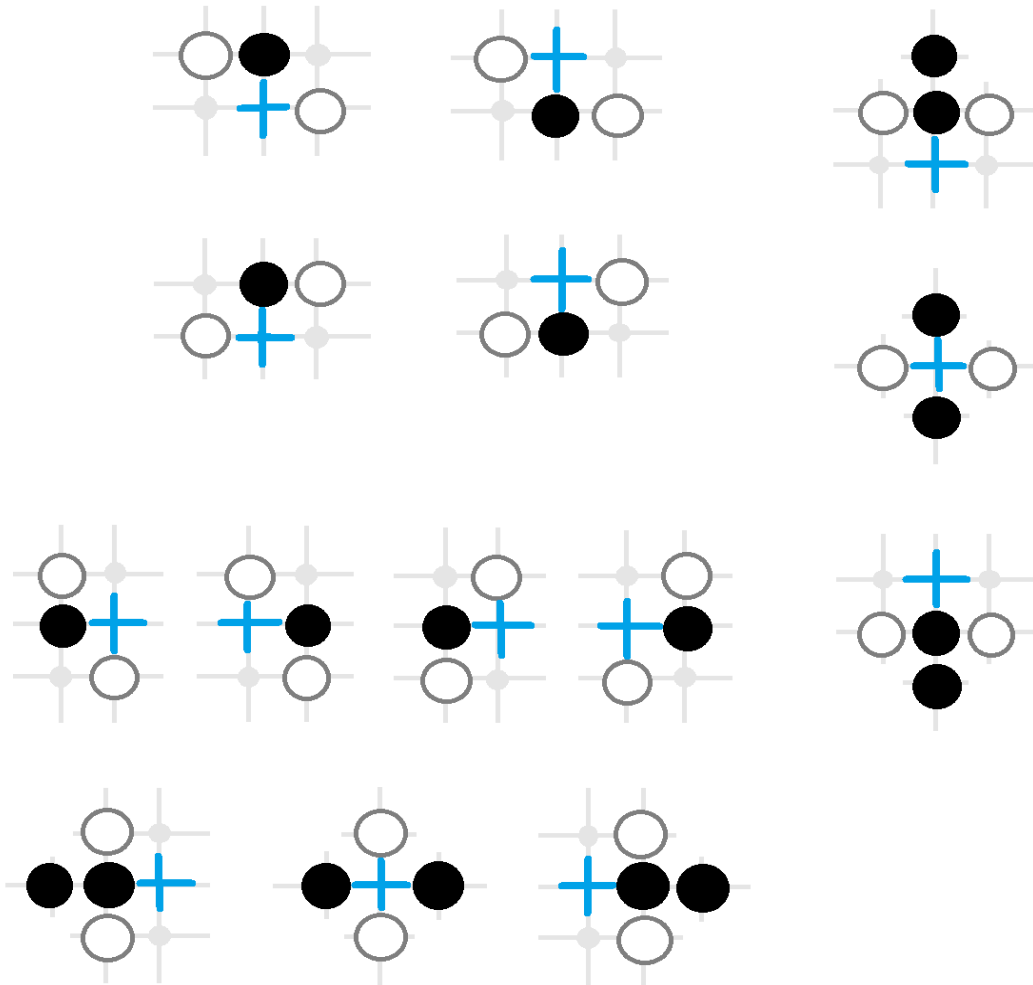
「逆に 好手形 も存在するのでは？」




「☞ 14パターンあるかも！」



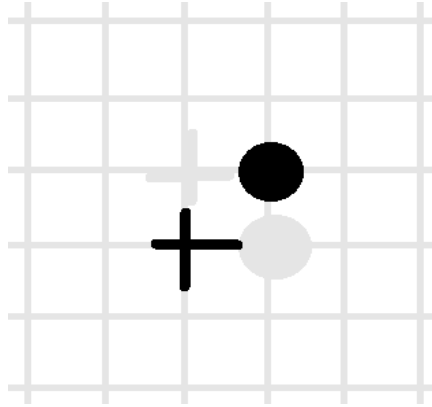
「将棋の駒の種類の数と同じねえ」



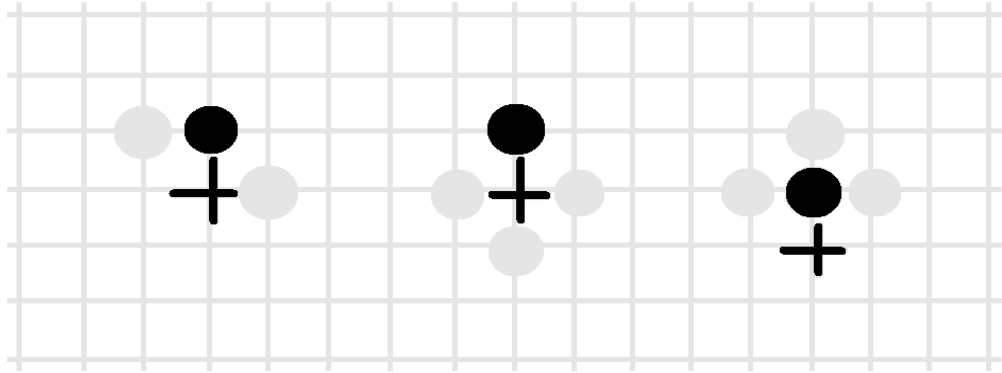
「 もう少し細かく言うと、
上図には 小さな灰色の丸 があるはずだぜ。
つまり 白石であってはいけない はずだぜ」



「好形と悪形の両方を満たしてしまう例外パターンって無いの？」



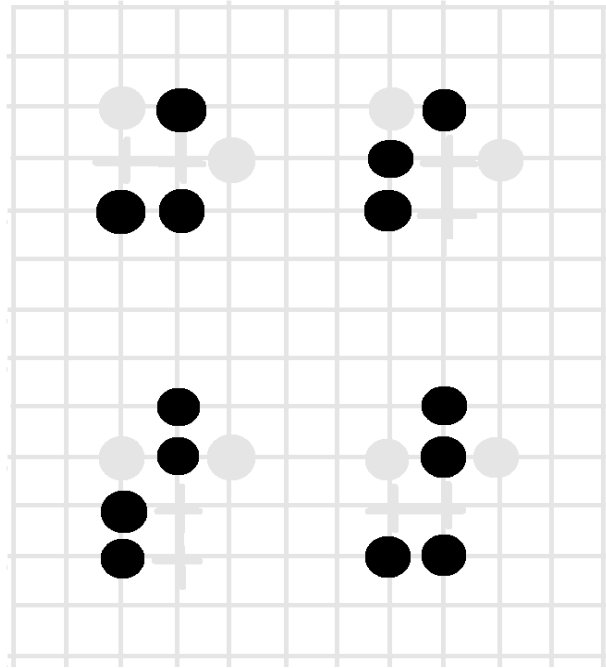
「☞ まず、悪形というのは、ナナメに置く、
という条件を満たすことでもあるんだぜ」



「☞ 次に 好形というのは、タテヨコ に置け、ということだぜ」



「ナナメと タテヨコの どちらも兼ねる図形を 教えるだぜ」



「☞ 同型を省けば、4種類 考えられるぜ」



「それは 好形なの？ 悪形なの？」



「囲碁強くないが この形で打たなかったら 切られるだろ。
打てば安心、だが 良い手かは知らないぜ」



「じゃあ 差し引き ゼロ で」



「ゼロ に調整するのも手間なんで、
プログラムの的には 無視する、ということにしようぜ？」

回転、対称で8パターンに展開

- expand_pattern3x3() 関数
- #define EXPAND_PATTERN_MAX (8*100)
 - パターンの最大数は100(8倍になるので)
- 90度回転を3回。線対称。また90度3回。

●○?	124	90度回転	?+●	401
+ + +	000	→	?+○	402
? ? ?	444		?+?	404

“?” は 4



「👉 講習会の レッスン10 で 山下さん何かやってるけど
アキ三角サカレガタチ理論 とアイデア被ってないの？」



「3×3パターンの中に アキ三角サカレガタチは
内包されてるな。
もう山下さんが全部がんばればいいのか」



「じゃあ レッスンをもう少し続けて 実装をパクればいいのか？」



「このPR文書を書き終えてから 読んでみて
使えそうなら 使おうぜ」

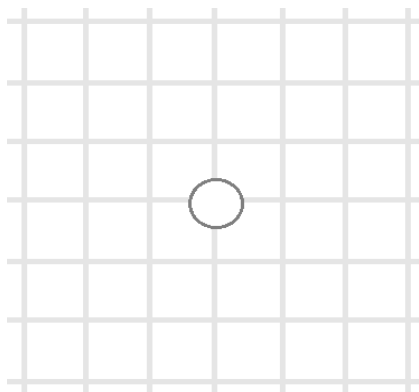
実用的に 実装 できるの？



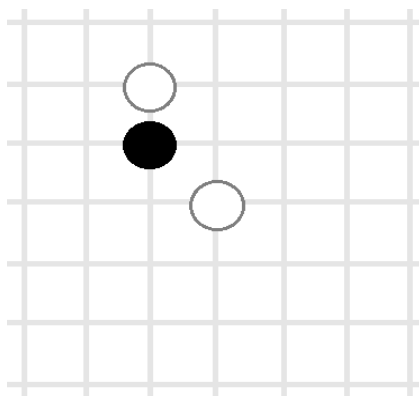
「盤面のどこに 好形 があるのか探したら、
時間がかかって しょうがない？」



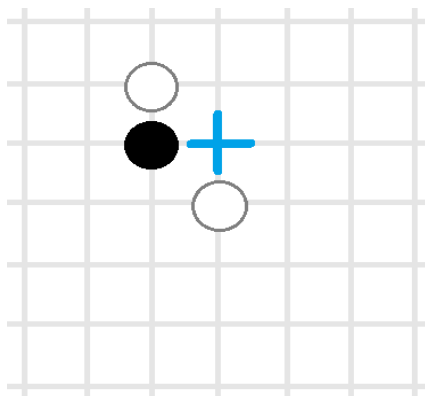
「好形 の場所は、相手が教えてくれるぜ」



「☞ 相手は どっかに 石を 置くだろ」



「☞ それが もしかすると 悪形の 1歩手前かもしれないだろ」





「☞ じゃあ 黒番 は、 好形 の1手を 記憶しろだぜ」



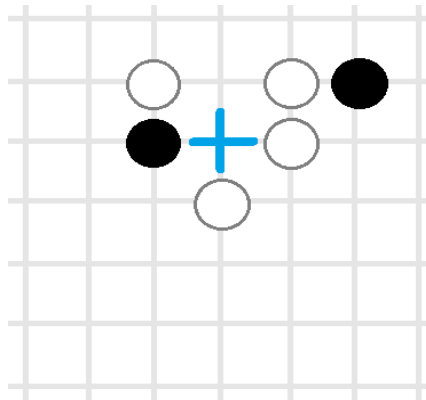
「 だったら 白番 は そんなところに 石を置かない気がするなあ」



「 別の向きで 好形 が有ったら 置くんじゃない？」



「 せっかく アキ三角サカレガタチ理論 を使うのだから、
貪欲に 好形の候補地に バンバン 置いていこうぜ？」



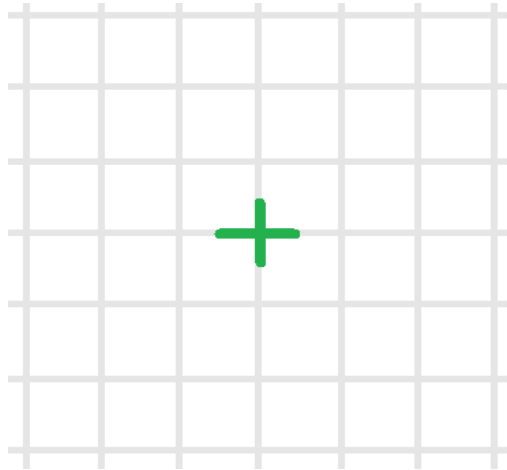
「☞ 一度 好形 と判定されて、そのあと 好形 で
なくなってるような ケースは どうするんだぜ？」



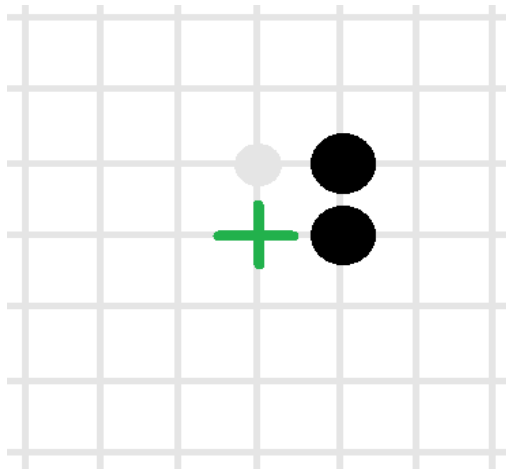
「 打つ前に 悪形チェックを 入れようぜ？」



「自分の手番！
モンテカルロ探索を やるのだから……」



「☞ 盤のどこに置くかなんか 考えるのではなく
乱数で 決めるぜ」



「☞ 置く前に その周囲を見て、
なんか 悪形 の 条件 を満たしていたら、
そこに 打つのは 止めて 別のところに 打てだぜ」



「お父んみたいな思想で 生きてるな」



「もう 悪形しかない! っとき どうすんの?」



「悪形 好形 関わらず、1局面につき

囲碁盤の交点の数 \times $2\pi e$

だけトライしたら

めんどくさいから

最後に石の置けたところに置くか、
それもなければ 負け にしようぜ?」

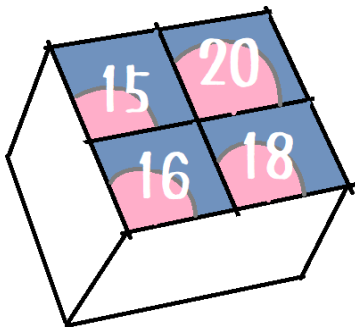
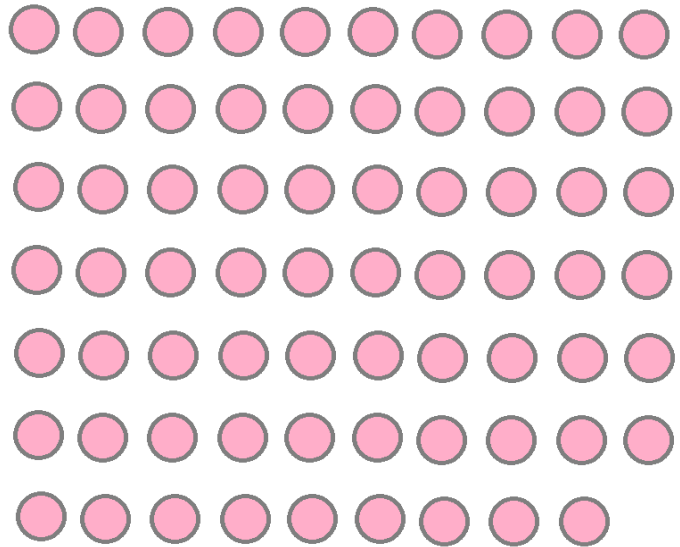
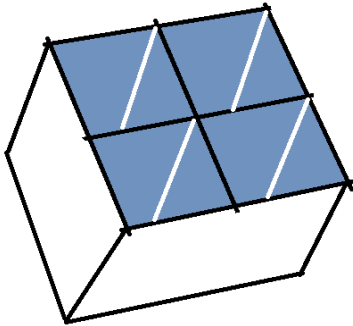


「考えるの 途中で止めてしまうの わらう」



「2 掛ける 円周率 掛ける ネイピア数 って 何なの?」

穴の数 4

 $2\pi e = \text{約} 17.08$ $4 \times 2\pi e = \text{約} 69$ 

「必ず どこかの箱には 入る玉が、
すべての箱に入るためには 何個 投げればいいのか？
というと 約 $2\pi e$ だと思ってるぜ。
ガウス分布を見ていて もしかして、と思った勘だぜ。自分調べ。
けっこう いい感じだぜ。
誰か 数学的に証明してくれだぜ」



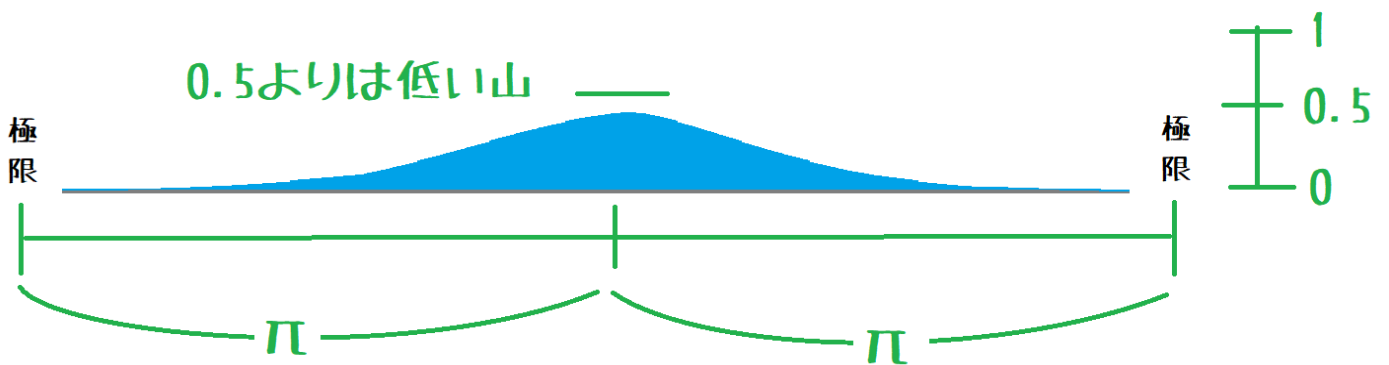
「そんなん調べる余裕があるなら
もっと 考えてくれだぜ！」



「☞ ガウス分布って こんなんだろ」



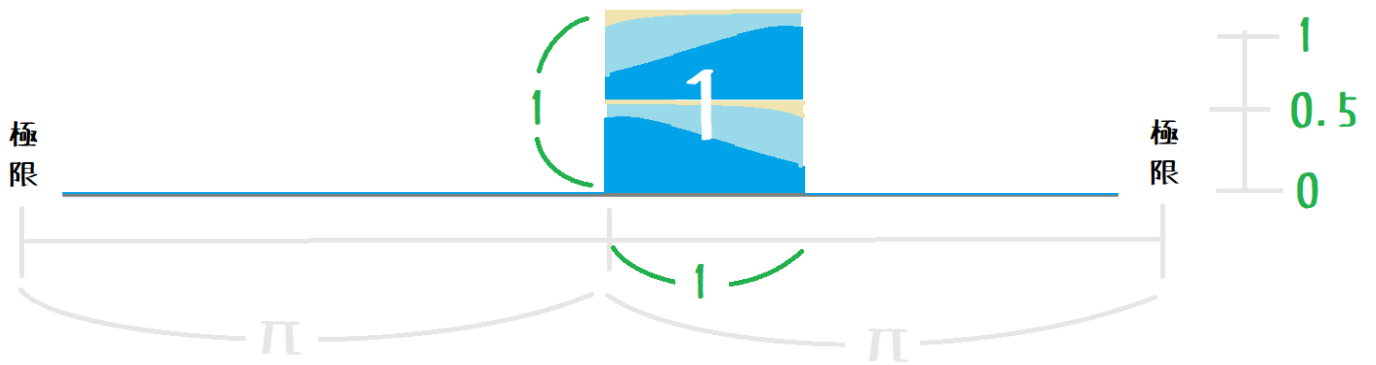
「聞いてないのに……」



「☞ サイズ感は わたしは こんなもんだと思ってるんだぜ」



「第13回UEC杯コンピュータ囲碁大会のPR文書に ページ数制限が無いのを いいことにお絵描きを始めてしまうの、わらう」



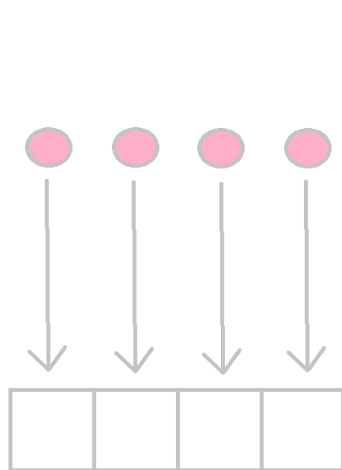
「☞ で、液体を集めると 1 になる……、というより
 事実は逆で 最初は 1 だったものを
 せつせと 横に引き伸ばしたのが ネイピア数の成り立ちだけ」



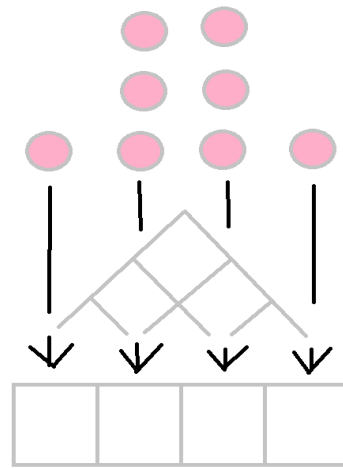
「何だぜ その プールの水が 1リットル みたいな図は？」



「 π (パイ) はあるけど、どこに e (ネイピア数) があんの？」



鳩の巣原理



パスカルの三角形



「☞ ここで思い出したいのは 鳩の巣原理と
 パスカルの三角形だけ」



「鳩の巣原理は 箱の数だけ ボールがあれば十分よね」



「1リットルの水が 1リットルの箱に 入るの、分かりやすいぜ」



「それに比べて パスカルの三角形では 底辺の箱に
ボールを全部入れるには、
箱の数が1つ増えるたびに ボールの数は2倍にしたいぜ」



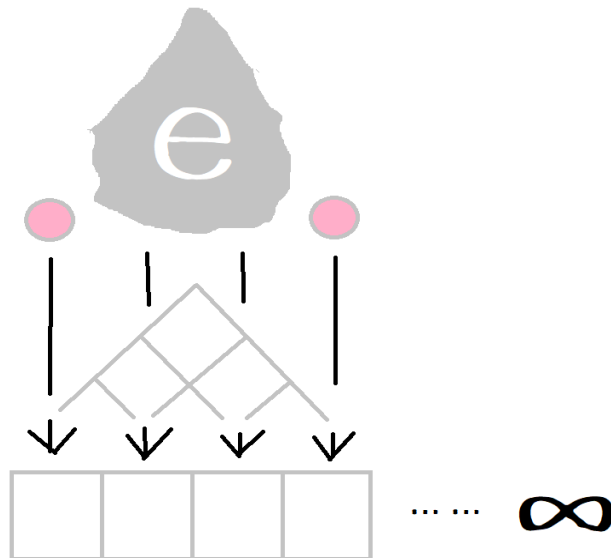
「それって どういう計算？」



「2の 箱の個数乗 ねえ」



「で、ネイピア数 というのは
箱の数が 無限大 に向かっていったときの
ボールの数 ぐらいに 思ってくれだぜ」





「☞ ネイピア数って 2.7 ちょい じゃないのか？
なんで ボールのところに居る？」



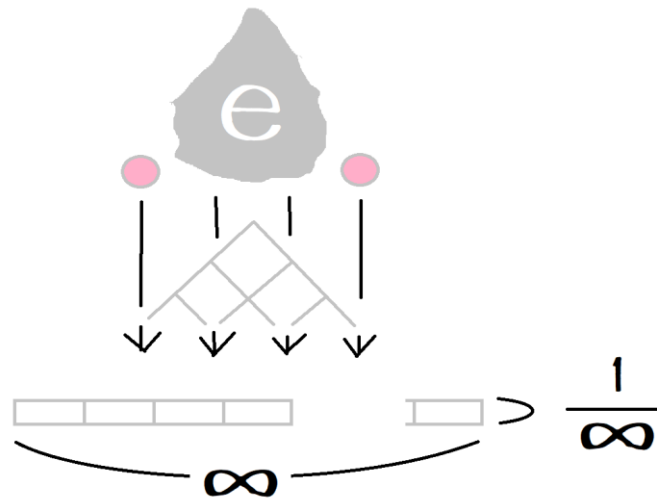
「小数点が無くなったと思ってください。
頭が 2718281828... といったことが分かっているだけの
無限に大きな数が、
横幅が無限に大きな床の上に居ると思ってください」



「そんなん 困るぜ」



「で、さらに 気分的な話しなんだが……」



「☞ ネイピア数の箱は 横幅は 無限大に向かっているが、
縦幅は 無限小 に向かっていると思うんだぜ」



「無限大は 数ではないから、
数みたいに 分母に置くのは 間違いみたいよ？」



「横幅と 縦幅を 掛け合わせれば 1 に戻ると思うんだぜ」



「箱が 1つになるのか？」

$$\text{width} \times \text{height} = \text{area}$$



$$e \times e = e^2$$



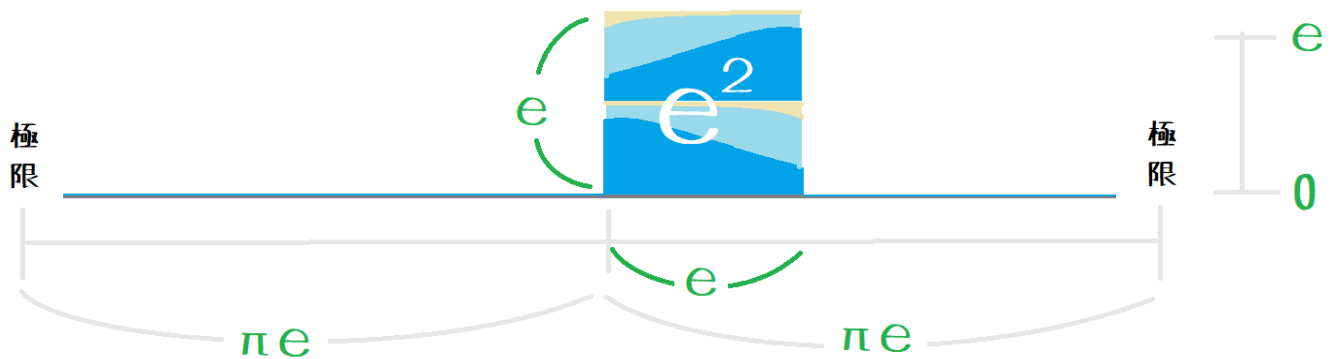
$$\infty \times \frac{1}{\infty} = 1$$



「👉 これだけ 分かれば プログラムで 乱数を使って試してみると
なんか まぐれで 超 いい感じに……」



「まぐれで 超 いい感じって 何なんだぜ！
ウソ数学じゃないか！ 解散 解散！」





「☞ で、わたしは プールの水が e の2乗 があると
思ったんだぜ」



「その水を ふちまければ $2\pi e$ の床を
水びだしに できると思ったのね」

```
C:\Github\random-pigeon-nest-hypothesis>python.exe main.py
[2022-01-28 20:24:00] Start!
[2022-01-28 20:24:08] N=1000 ok (Failure count=0 Success rate=1.0)
[2022-01-28 20:24:32] N=2000 ok (Failure count=0 Success rate=1.0)
[2022-01-28 20:25:15] N=3000 ok (Failure count=0 Success rate=1.0)
[2022-01-28 20:26:07] N=4000 ok (Failure count=0 Success rate=1.0)
[2022-01-28 20:27:26] N=5000 ok (Failure count=0 Success rate=1.0)
[2022-01-28 20:28:55] N=6000 ok (Failure count=0 Success rate=1.0)
[2022-01-28 20:30:30] N=7000 ok (Failure count=0 Success rate=1.0)
[2022-01-28 20:32:16] N=8000 ok (Failure count=0 Success rate=1.0)
[2022-01-28 20:34:46] N=9000 ok (Failure count=0 Success rate=1.0)
[2022-01-28 20:37:34] N=10000 ok (Failure count=0 Success rate=1.0)
[2022-01-28 20:40:32] N=11000 ok (Failure count=0 Success rate=1.0)
[2022-01-28 20:43:38] N=12000 ok (Failure count=0 Success rate=1.0)
[2022-01-28 20:46:52] N=13000 ok (Failure count=0 Success rate=1.0)
[2022-01-28 20:48:39] N=13539 Fail. There are 1 missing balls. 7.386069872220991e-05 (Failure count=1 Success rate=0.9999261393012778)
[2022-01-28 20:50:10] N=14000 ok (Failure count=1 Success rate=0.9999285714285714)
[2022-01-28 20:53:38] N=15000 ok (Failure count=1 Success rate=0.9999333333333333)
[2022-01-28 20:57:18] N=16000 ok (Failure count=1 Success rate=0.9999375)
[2022-01-28 21:02:03] N=17000 ok (Failure count=1 Success rate=0.9999411764705882)
[2022-01-28 21:07:30] N=18000 ok (Failure count=1 Success rate=0.9999444444444444)
[2022-01-28 21:13:03] N=19000 ok (Failure count=1 Success rate=0.9999473684210526)
[2022-01-28 21:18:43] N=20000 ok (Failure count=1 Success rate=0.99995)
[2022-01-28 21:20:50] N=20363 Fail. There are 1 missing balls. 4.9108677503314834e-05 (Failure count=2 Success rate=0.9999017826449934)
[2022-01-28 21:24:34] N=21000 ok (Failure count=2 Success rate=0.9999047619047619)
[2022-01-28 21:26:15] N=21287 Fail. There are 1 missing balls. 4.697702823319397e-05 (Failure count=3 Success rate=0.9998590689153004)
[2022-01-28 21:30:32] N=22000 ok (Failure count=3 Success rate=0.9998636363636364)
[2022-01-28 21:31:55] N=22224 Fail. There are 1 missing balls. 4.499640028797696e-05 (Failure count=4 Success rate=0.999820014398848)
[2022-01-28 21:33:10] N=22428 Fail. There are 1 missing balls. 4.4587123238808635e-05 (Failure count=5 Success rate=0.999777064383806)
[2022-01-28 21:36:42] N=23000 ok (Failure count=5 Success rate=0.9997826086956522)
[2022-01-28 21:40:10] N=23548 Fail. There are 1 missing balls. 4.246645150331238e-05 (Failure count=6 Success rate=0.9997452012909801)
[2022-01-28 21:43:00] N=24000 ok (Failure count=6 Success rate=0.99975)
```



「☞ 水びだしに できてないところもあるわよ？」



「☞ 超 水びだしに できてるぜ！
2万3千個のボールを 2万3千 掛ける 17.08 回
投げて
ボール1個 足りないだけとか 超 ギリギリを攻めてるぜ！
ランダムの鳩の巣仮説 だぜ！」



「証明してくれだぜ！
いつになったら 定理になるんだぜ」

>>> My dad is a triangle,
and I'm a square !?