CORDIC Idea Backtracking

20160129

* Backtracking

    when to backtrack

    how to backtrack?

* Cut

* Coarse - Fine Approach

1 Back Tracking

ⓐ Brute force, traditional backtracking

ⓑ Heuristics

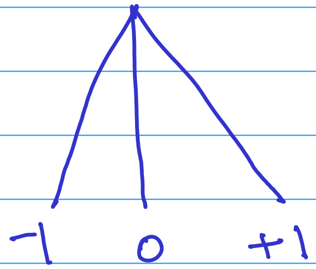
avoid "dense" angle?

0, 45°, 90°, 135° .....

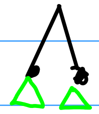
leading ⓪ or ①

② cut —— ternary tree

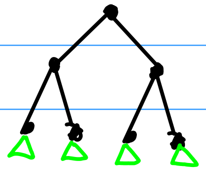
after enough iterations, constant scaling problem vanishes



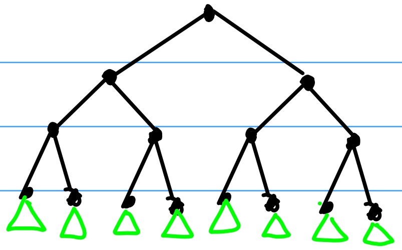
What is backtracking?



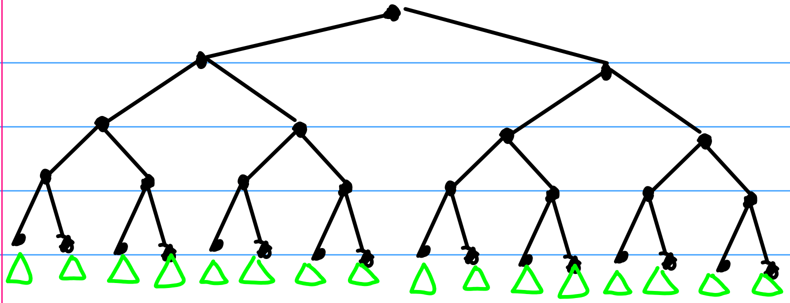
- Original CORDIC algorithm
  — basically DFS.



- Overlapping area (dense angle region)
  - try to search other subtrees (descendants of sibling nodes)
  - with the aim of reducing total number of iteration enhancing the precision



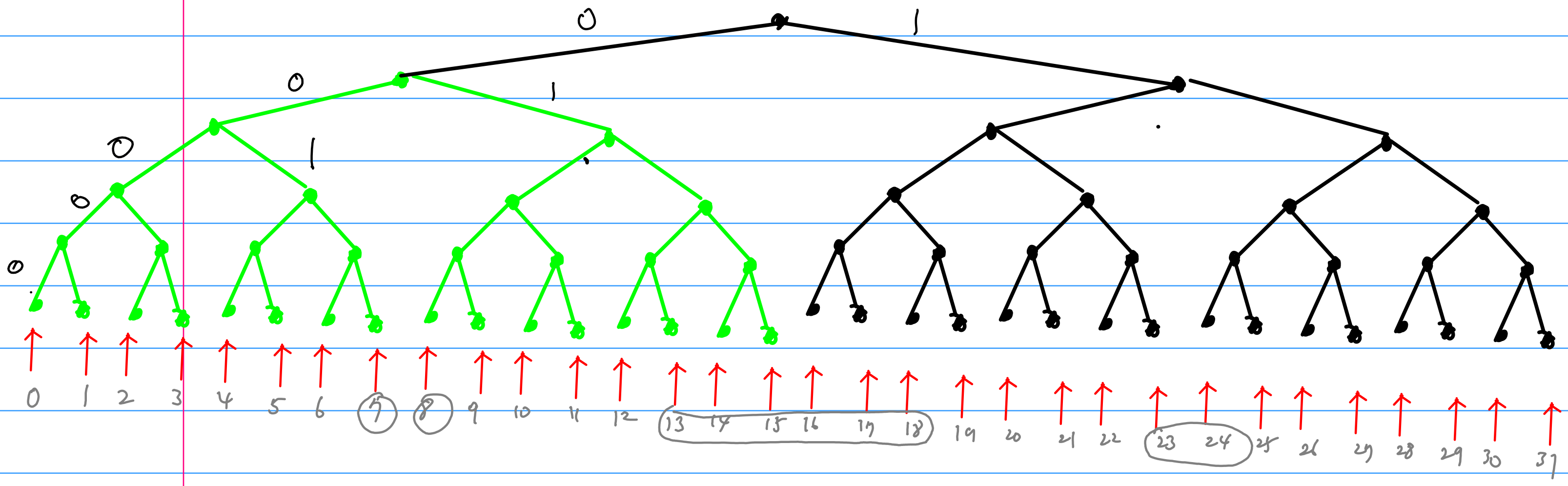
  - may suitable for the initial stage of iteration (first 1~5 steps of iteration)



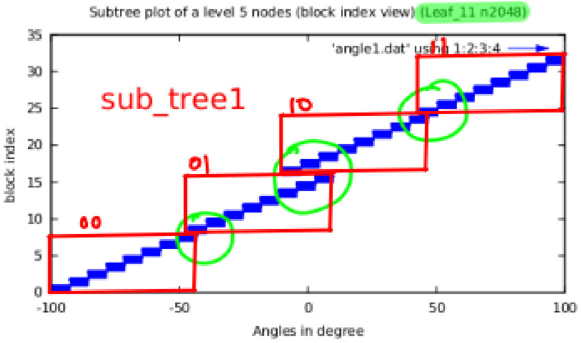
⑥ must check the result of finding optimal solution (BFS)

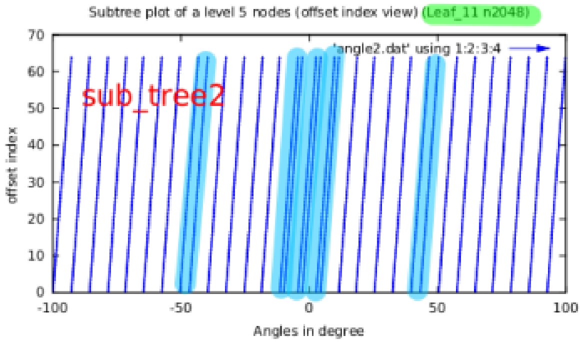
Angle Ordering

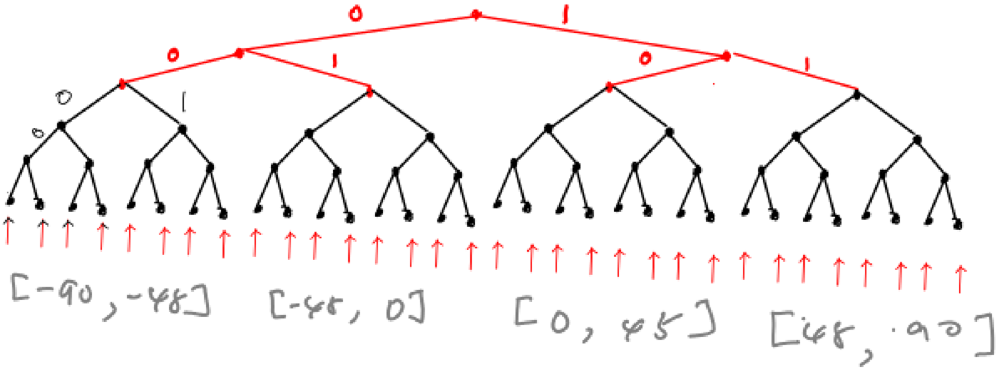
angle Index Ordering



Sorted angle Ordering



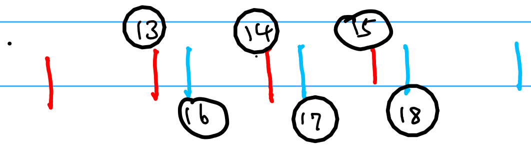


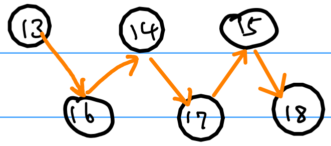


approximate angle ranges

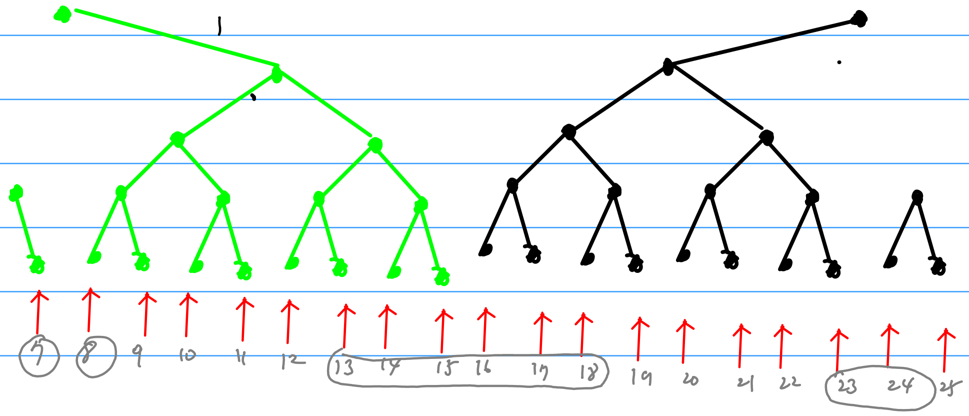
angle index ordering

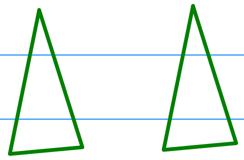
00000	0
00001	1
00010	2
00011	3
00100	4
00101	5
00110	6
00111	7
01000	8
01001	9
01010	10
01011	11
01100	12
01101	13
01110	14
01111	15
10000	16
10001	17
10010	18
10011	19
10100	20
10101	21
10110	22
10111	23
11000	24
11001	25
11010	26
11011	27
11100	28
11101	29
11110	30
11111	31

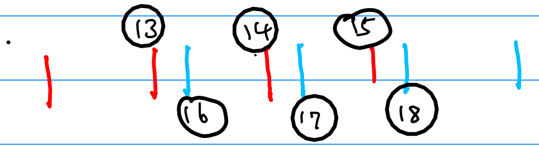


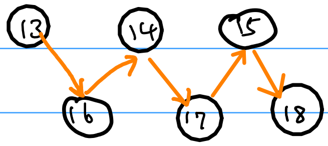


Sorted angle ordering



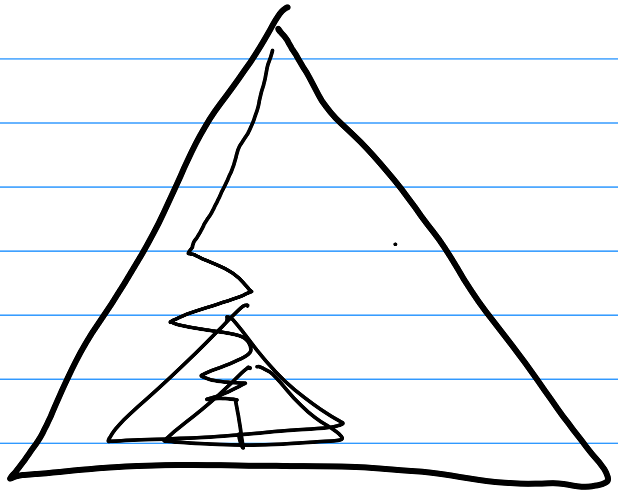






Sorted angle ordering

Backtracking

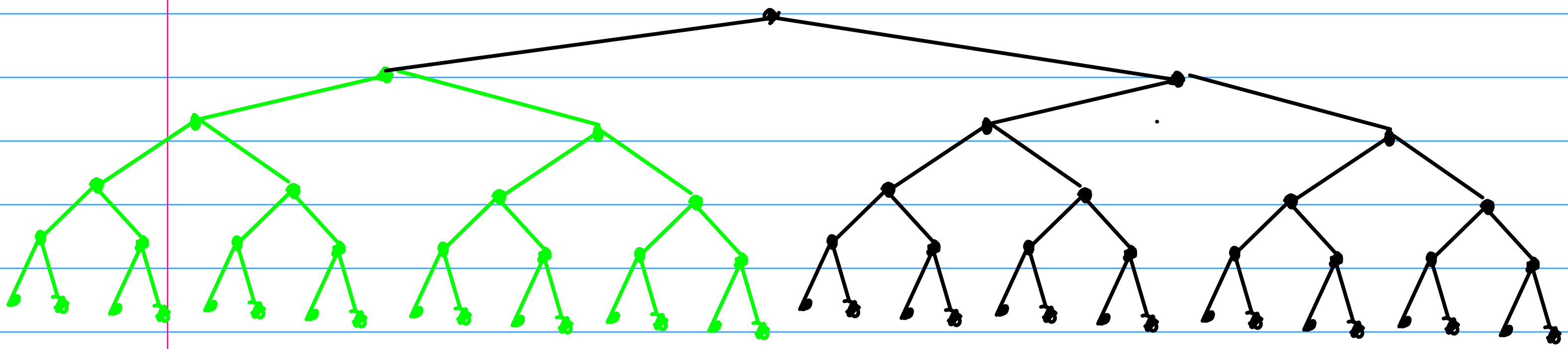


try some other neighbor subtrees in sorted angle ordering

# of backtracked levels are different ...

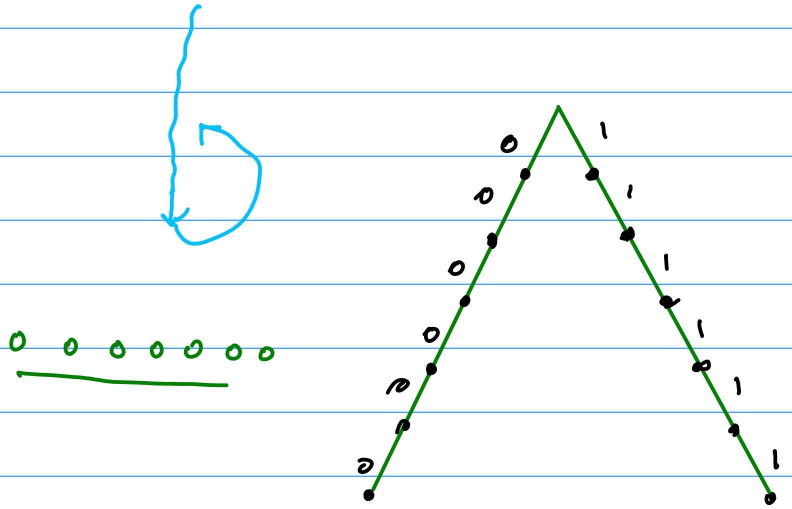
Can't search all possible children

may the combination of DFS and BFS hopefully



when to backtrack

But when to backtrack?

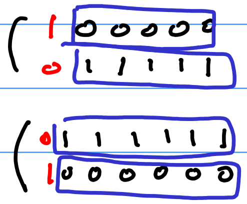


① consecutive 1's or 0's in the angle path.

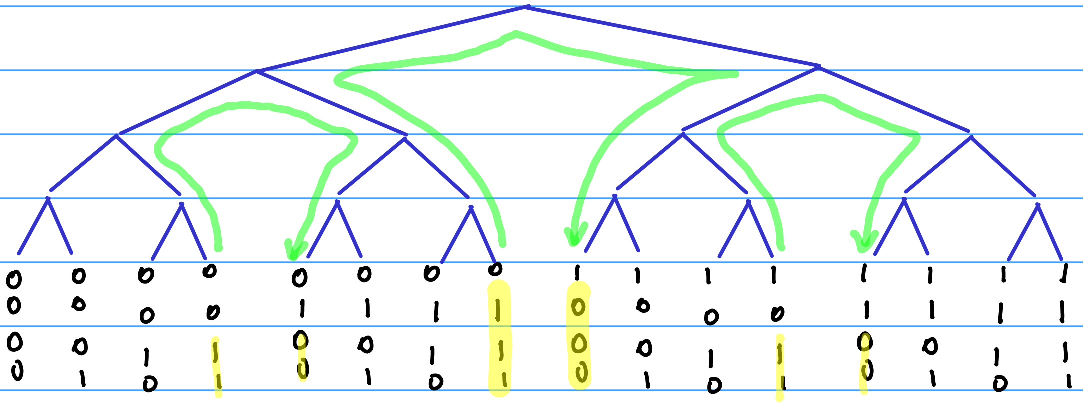
$\longrightarrow$ increasing level

i	$i+1$	$i+2$	$i+3$	$i+4$
0	0	0	0	0
$+\theta_i$	$+\theta_{i+1}$	$+\theta_{i+2}$	$+\theta_{i+3}$	$+\theta_{i+4}$

during all the consecutive 5 iterations, angles were always added.



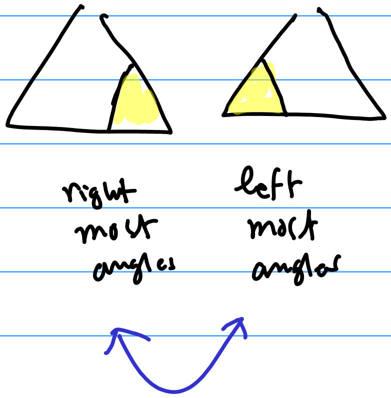
how to backtrack?



0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

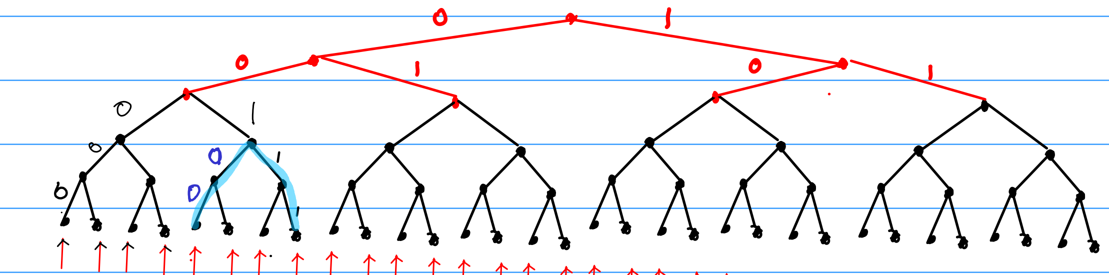
0	0	1	1
1	1	0	0
1	1	0	0
1	1	0	0
0	1	0	1
14	15	16	17

w
overlapped



any difference?

need to check!

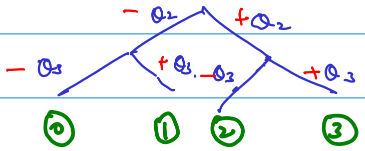


0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

θ_1

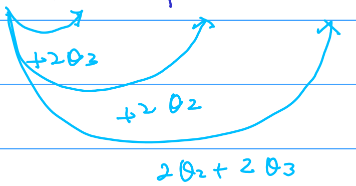
$\uparrow \theta_2 \qquad -\theta_2 \qquad +\theta_2'$

$\theta_3 \qquad -\theta_3 \qquad +\theta_3'$

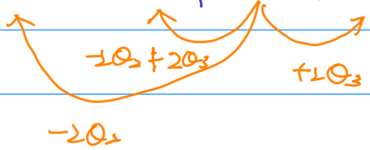


θ_2	−	−	+	+
θ_3	−	+	−	+

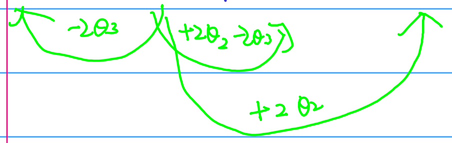
	⓿	①	②	③
θ_2	−	−	+	+
θ_3	−	+	−	+



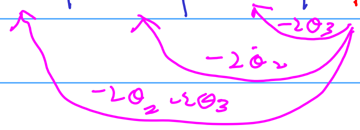
	⓪	①	❷	③
θ_2	−	−	+	+
θ_3	−	+	−	+

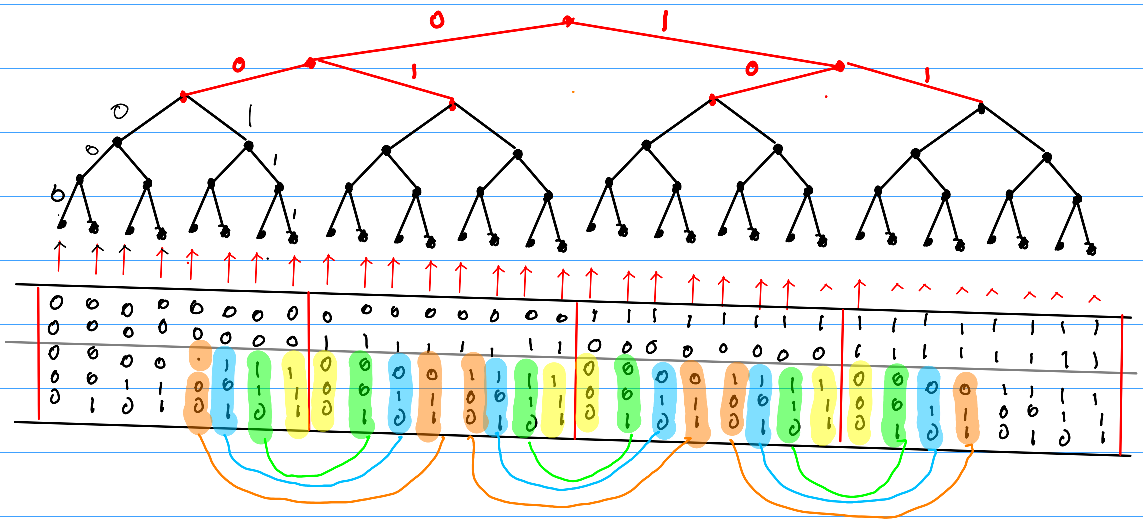


	⓪	❶	②	③
θ_2	−	−	+	+
θ_3	−	+	−	+



	⓪	①	②	❸
θ_2	−	−	+	+
θ_3	−	+	−	+

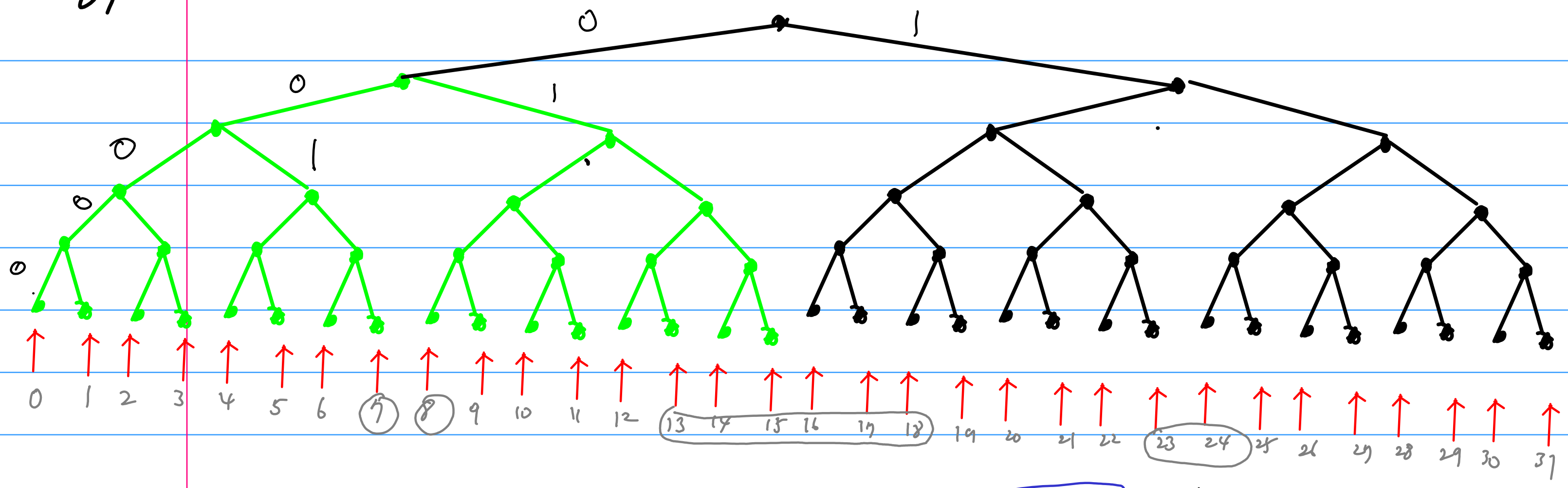


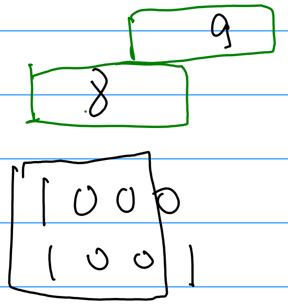


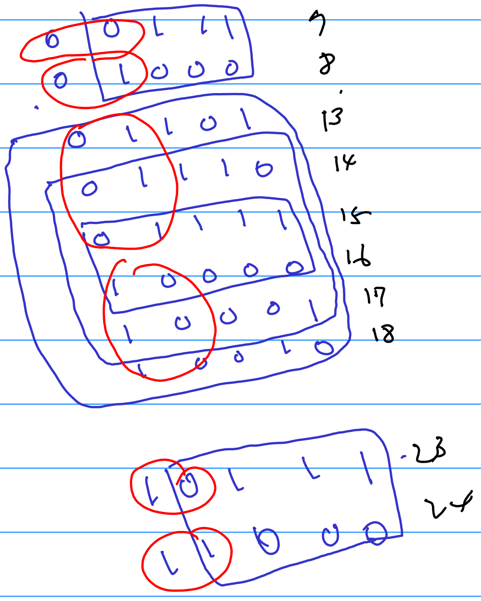
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	0	0	0	.	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

blocks 7,8
blocks 13,14,15,16,17,18
blocks 23,24

$\frac{16}{24}$





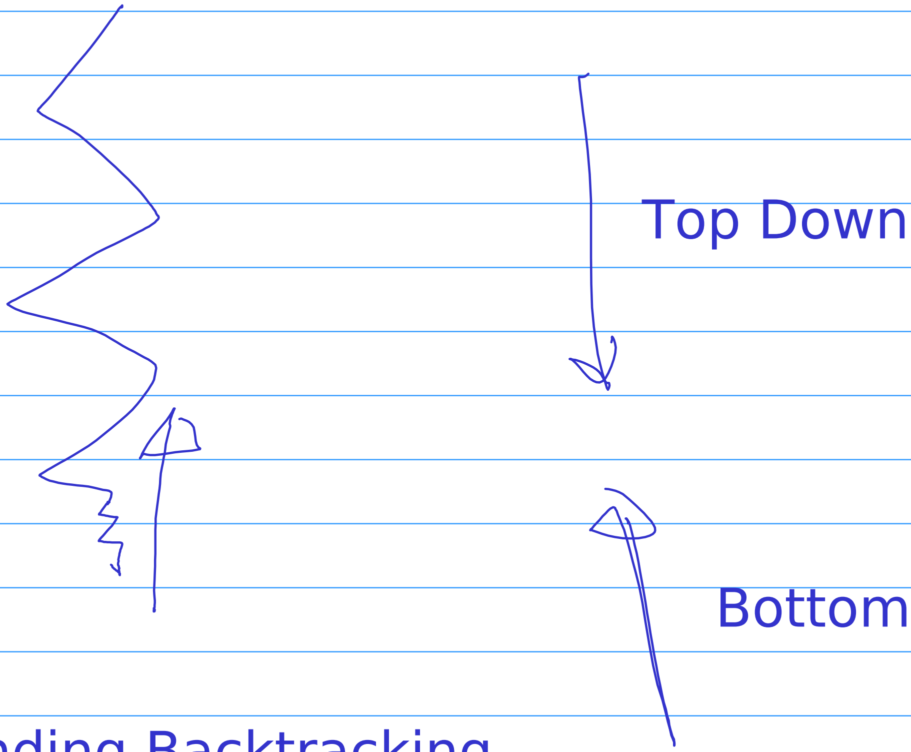


blocks 7,8
blocks 13,14,15,16,17,18
blocks 23,24

☆ round off errors
offset binary effect
always increasing ?

angle index ordering different
sorted angle ordering

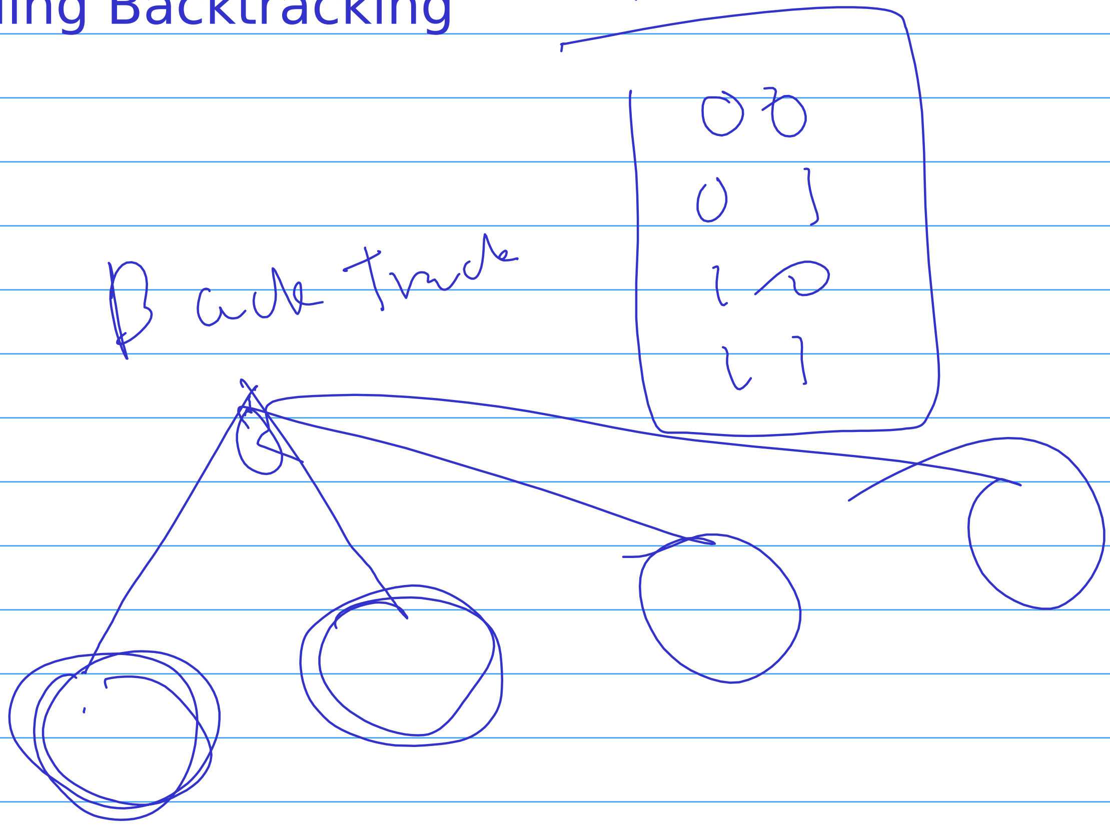
Sometimes it may be better
try some successors/ predecessors
in sorted angle ordering
than going deeper



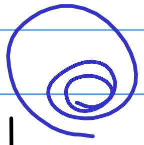
Top Down

Bottom Up

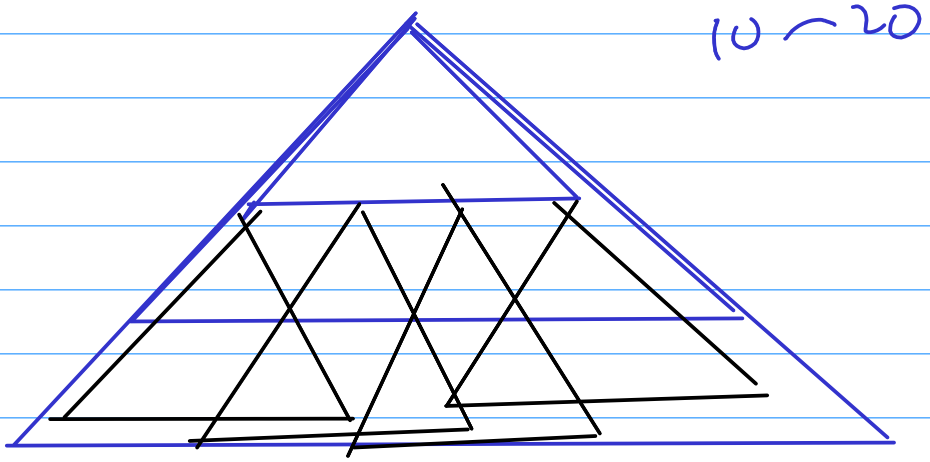
Extending Backtracking



0 0
0 1
1 0
1 1



Need to check the angle ranges of some nodes that are ihe initial iterations of 4 or 5



what angles CORDIC can represents?

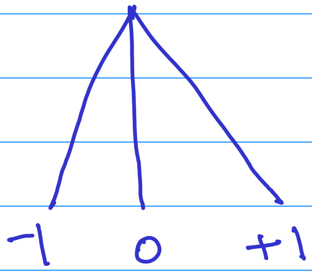
dense area (overlapped region) more accurate / more precise
sparse area (non-overlapped region) less accurate / less precise

if CORDIC is used what is the error / precision

for a given angle, what does CORDIC give?
and is it a really best (optimal) angles of CORDIC?
or can we find other CORDIC angle representations? (uniqueness)

④ ternary tree

after enough iterations,
constant scaling
problem vanishes



if "early cut" method is used, the error looks like insignificant?

if CORDIC gives an angle representation within a specified error range,
then break CORDIC iteration, using the current value as final

constant problem ?
--> it doesn't matter after initial iteration
because Ki of each iteration approaches to 1.

So actually no scaling problem after a few initial iterations
--> some literature mentions about
scaling free cordic algorithms, so let's find and read throught....

Virtually scaling-free adaptive CORDIC rotator

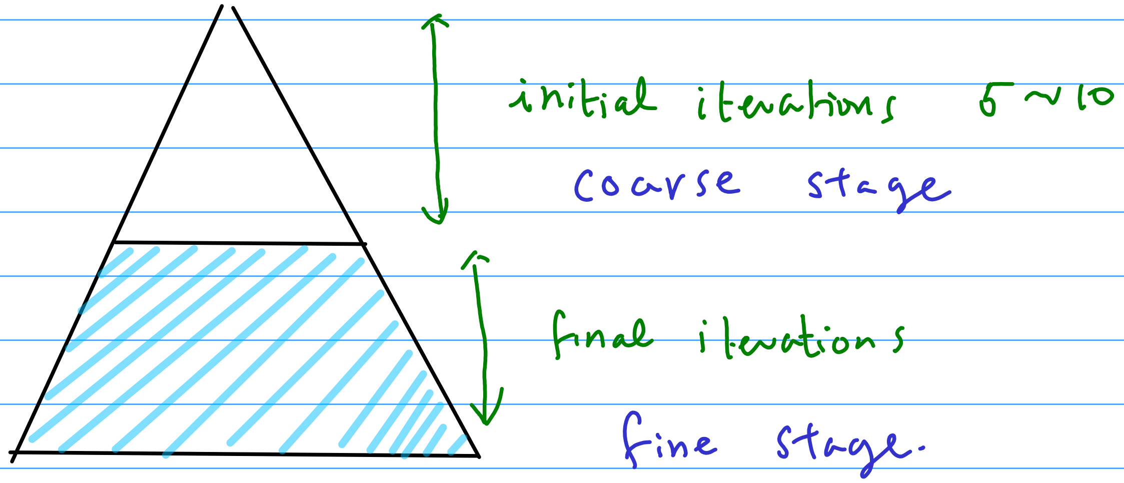
K. Maharatna, A. Troya, S. Banerjee, E. Grass
IEE Proc 2004

Coarse - Fine Approach

COARSE - FINE approach
depending on the Ki terms
after certain iteration product of Ki remains constant

Multiple CORDIC Core
each start with different angle shift

Parallel + Pipeline + Something about precision?


initial iterations 5 ~ 10
coarse stage
final iterations
fine stage.