PIBU
PINEAPPLE BUILDER AND TESTER

Again THANKS TO SCHAAKE AND TURNER. (I have somewhat rewritten the maths here for clarification but the "hard part" comes from them, not from me)
(you do need to read *THE BRAIDER* collection)

Abandon all hope of using this one if you are not "very clear" about what is a Pineapple and what IS NOT a pineapple.

If you belong to the crowd that mistakenly call Pineapple ANY knots with NESTED BIGHT then pass you way please, you are only losing your time.
Before that you have to read :

BRAIDING – Standard Herringbone Pineapple Knots (200+ pages)
by A.G. SCHAAKE , J.C. TURNER , D.A.SEDGWICK

At least read Turkshead_15 (and 14) to get " a wee bit notion" as you will not be able to conceive your knot and then enter specification (A , Total LEAD (TL) , Number of BIGHT NEST (BN)) for the program to works on.

Those are the knots **this program will BUILT THE SET OF COMPONENT THK and TEST if it is POSSIBLE OR NOT POSSIBLE** (2 general cases of impossibility exist : PARITY conflict between A and TL or GDC conflict inside at least one SET of COMPONENT THK)

Those are **STANDARD HERRINGBONE PINEAPPLE KNOTS** (S-H-P-K) better known as PINEAPPLE (plus a lot of mongrels knots thrown with them by the not curious enough about what they have been "handed down" !) for short.
They are an assembly of component THK, "true" THK that is.
(for clarification read my pre-*THE BRAIDER* features : **THK or NOT THK** and **THK and MATHEMATICS**)

This program can do (if you know how to erase just a few command line) SEMI-STANDARD H-P Knots that is knots where one or several component THK is not a true THK because it is not a single strand THK.

You will need to enter only 3 numbers

- A (Number of **PASS** ; that is number of COMPONENT THK)
- B* (Number of BIGHT NEST ON EACH RIM)
- Total number of LEAD in the finished knot (that is sum total of the component THK leads)

For that small price you will get
--- the number of component THK in each of the TWO SET (two sets of THK)
--- The NUMBER OF COMPONENT THK IN EACH SET,
--- THE LEAD AND BIGHT of the COMPONENT THK IS A SET

or one short answer…… IMPOSSIBLE.

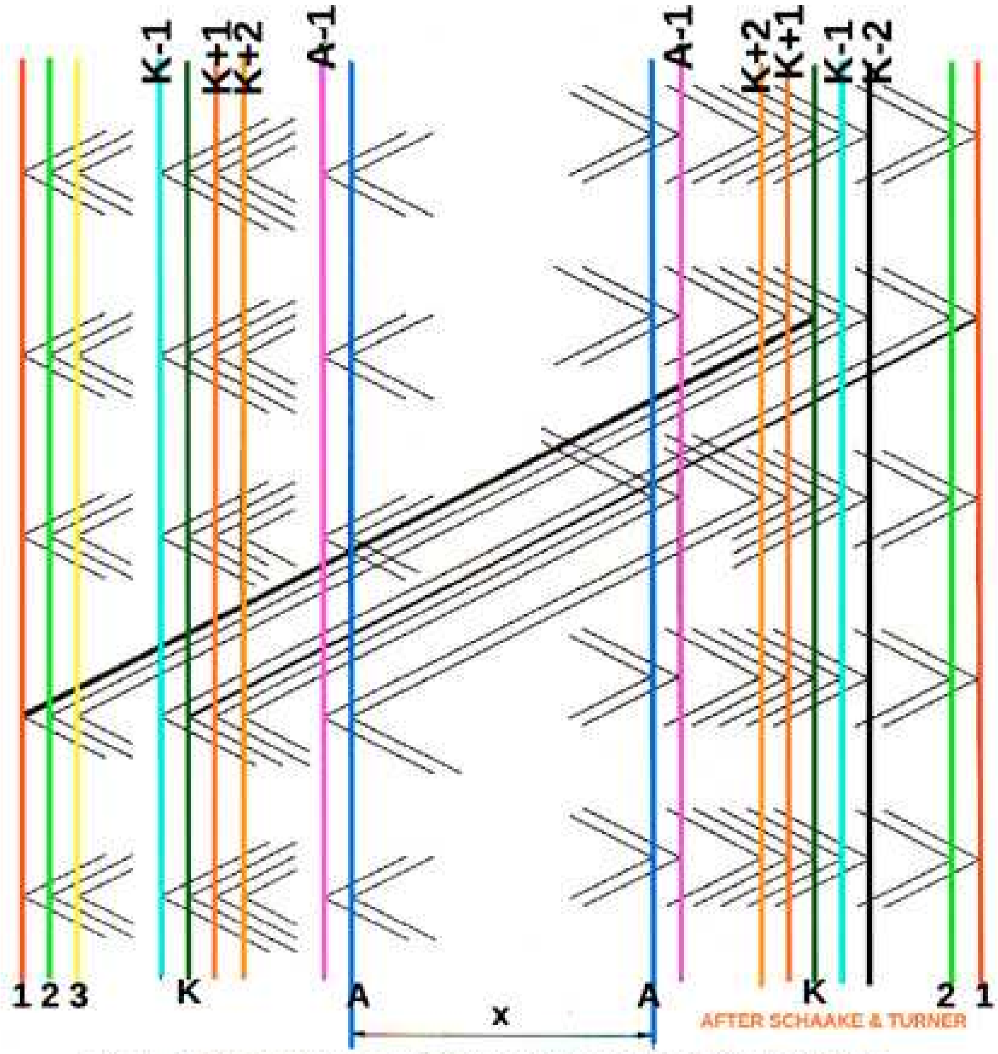

Fig. 3 — Half-cycles between left Bight-boundaries and right Bight-boundaries.

ANY S-H-P-K belong to the REGULAR NESTED CYLINDRICAL (BRAID for S & K) (but for me they are NOT BRAID - see THK ARE NOT BRAIDS a pre-*THE BRAIDER* feature of mine) **CLASS**

This **CLASS enclose much more than S-H-P-K** . (see Turkshead-15 page)

ANY S-H-P-K has NESTED BIGHT but any knot with NESTED BIGHT IS NOT A PINEAPPLE.

There is a strict logic in using 1, 2 , 3…..A on each side : you cannot do what you want but what you must to get a “real” S-H-P-K.
This is not the place to instruct you about that. *** see the annex at the end
You should must know that before going about “conceiving” S-H-P-K and testing their “building”.

ANY S-H-P-K is an interlocking (do humour me and do not say interbraided !) of component “true” THK.

ALL THESE (in the two SET) **COMPONENT** THK HAVE THE **SAME NUMBER OF BIGHT : equal** to **B*** which is the Number of **BIGHT NEST** too.

ALL COMPONENT HAVE AN **ODD NUMBER OF LEAD**

They are distributed in **TWO SET : 'a1' and 'a2'** (one of those SET can be "empty" and all the component then belong to the unique "non empty" SET with ***AT LEAST*** 2 members)

BETWEEN THE TWO SET THERE IS A DIFFERENCE IN THE ODD NUMBER OF LEAD OF THE COMPONENT THK (difference is TWO)

One SET has **(2m – 1) LEAD** and the other one has **(2m + 1) LEAD (this is the 's' that you will use in the algorithm tables**

Something more : that is at the root of the algorithm tables one can built :
--- along an Half-Period (H-P) considered in a *FINISHED* S-H-P-K the crossings can sorted into **SET OF *ADJACENT* CROSSINGS OF *SAME TYPE*** (Over or Under)

The number of those SET OF CROSSINGS is denoted by 's' and the value of 's' is the Number of LEAD in the considered component THK being scrutinised in the algorithm tables.
For more on the algorithm tables you will have to read the book.
I will not give them free as this would be well beyond "fair quote" and contrary to my objective that is to help Dr John TURNER making this work known :
Contact Dr TURNER (via my What is new ? page and be a good buyer **: it is "at cost" and not at "commercial price"**)

My PROGRAMS **PINAPL** AND **PINAPL2** work those tables for you if you are "brain-engaged" and know the inside nature of a "true" **PIN**EA**P**P**L**E. Good luch decrypting the HP-RPL code and re-thinking the whole of the table building and using : believe me it is quicker to buy the book ! No hard feelings.

A the number of **PASS** is equal to the SUM of the COMPONENT THK that belong to one of the TWO SET hence **A = a1 + a2**
'a1' number of component THK in set 1 and 'a2' number of component in set 2.

Now about **PARITY**
'A' is 'ODD' IF and ONLY IF 'a1' and 'a2' are of OPPOSITE PARITY (that is so easily seen, as the nose in the middle of the face as goes the French saying, that I hope that you will not feel insulted if I give the explanation :
ODD added to ODD cannot be else than EVEN
Think EVEN as being (EVEN –1) (or (EVEN +1)
Adding (3 ways)
(EVEN –1) + (EVEN –1) = 2*EVEN – 2 = that is EVEN
(EVEN –1) + (EVEN +1) = 2 * EVEN = again EVEN
(EVEN +1) + (EVEN +1) = 2 * EVEN + 2 = EVEN too

'A' is 'EVEN' IF and IF ONLY 'a1' and 'a2' are of SAME PARITY (I will not risk insulting you again and give the reason why)

....

Total NUMBER OF LEAD (will be denoted FL) IN THE *FINISHED* S-H-P-K is
FL = a1 (2m – 1) + a2 (2m + 1) == **2m * A + (a2 – a1)**
I will leave you do this easy exercise, will be good for your neurons ! ;-)

Using this **2m * A + (a2 – a1)**

Will get us **m = (FL – (a2 – a1)) / 2 * A**

We also have

(FL) modulo A == (a2 –a1) modulo A

implying
(FL) modulo A = a2 – a1 for **1 <= a1 <= a2**
or
(FL) modulo A = A + (a2 –a1) for **a1 > a2 >= 0**

as **a2 = A – a1** we can write again

(FL) modulo A = A – 2 * a1 for **1 <= a1 <= a2**
or
(FL) modulo A = 2 * A – 2 * a1 for **a1 > a2 >= 0**

So it follows that

When **1 <= a1 <= a2**

a1 = (A - (FL) modulo A) / 2

a2 = A – a1 =(A +(FL) modulo A) / 2

2m = (FL - (FL) modulo A) / A
Let us call it that GROUP A

When **a1 > a2 >= 0**

a1 = (2 *A - (FL) modulo A) / 2

a2 = A – a1 = (FL) modulo A / 2

2m = ((FL - (FL) modulo A) / A) + 1

Let us call it that GROUP B
-

A and FL of OPPOSITE PARITY = IMPOSSIBLE S-H-P-K

Remain two cases (with each 2 sub-cases)

A and FL BOTH ODD

A and FL BOTH EVEN

There are sub-cases inside each of those two cases

A and FL BOTH ODD
a1 is an INTEGER (cannot be anything else !)

The equation to be computed are in two series !

(FL) modulo A is also **ODD**
GROUP A

(FL) modulo A is **EVEN**
GROUP B

A and FL BOTH EVEN

(FL) modulo A is EVEN

2m = (FL – (a2 – a1)) / A is EVEN

equations to be computed are in two series !

-
when
2m = FL -(FL) modulo A) / A
is **EVEN**
GROUP A

when
2m = (FL -(FL) modulo A) / A) + 1
is **EVEN**
GROUP B

This will give you the values of number of ways to do the knot, x, a1 , a2, 2m.

You entered
NUMBER OF PASS
NUMBER OF LEAD IN FINISHED S-H-P-K,
NUMBER OF BIGHT NESTS (that is also the number of BIGHT in each component THK whatever the SET it is belonging to :

You will get out of that if the knot is POSSIBLE

- **a1 which is the number of component THK is SET 1**
- **a2 which is the number of component THK is SET 2**

Then number of BIGHT you already know as you entered it.

The Number of LEAD will be calculated for each component tHK in each set using 2m and adding or subtracting '1'

The results are put (with labels) on the STACK.

Just one more 'funny' point :

'A' gives you the number of different ways the knot can be done.
It is simple the permutation of A object in A places.
This is given by " FACTORIAL "
Written A !
That is calculated as
A * (A – 1) * (A – 2) * …….* 1

A= 5 5 * 4 * 3 * 2 * 1 = 120 different ways to go about the job. Even that is put on the STACK

4 ! 4 * 3 * 2 = 24
3 ! let us do it with LETTERS
A B C B A C C A B
A C B B C A C B A
3 'object' that permute (musical chairs) in 3 'place'
3 * 2 * 1 == 3 * 2 = 6 there is no 7^{th}

Note that if you don't consider the "place' but just the set the 3 'object' taken by helping of 3 can only leads to one way of doing that (putting back the items that were pulled each time)

While (still) without considering the place (or the order in which you 'pull them from the bag ' with 4 items (A, B, C, D) you can have
A B C (count as only one with C B A, B A C , C A B , B C A, A C B)
A B D A D C B D C
4 different ways of having 3 items out of 4
This will help you if you have only 4 colours of cordage and want to use only 3, then you know how many choices you have ! Of course there are formulas that make the job easier say 3 out of 8 colours ?
http://old.andrews.edu/~calkins/math/webtexts/prod02.htm

try to use the **PIBU** PGR with those entries

BIGHT NEST number	FL (final total LEAD)	A number of PASS
5	41	5
4	15	34
4	41	3
4	29	5
4	31	5

and see what you get.

ANNEXE

Have several good look at the illustration given.

With the results of the program run you have

a1 SET which have **s = 2m - 1**

a2 SET which have **s = 2m + 1** (only when a2 is not an “empty set “ with value zero)

When using tables this means using two different algorithm tables when building a S-H-P-K , one table with one ‘s’ and the other with another ‘s’.
To avoid GDC conflict the two ‘s’ have to be each prime relatively to B*

Lets us have q quick look at the way the component THK are taking their place in the S-H-P-K being built.

In the illustration under only ODD H-P are shown going (mandrel frame of reference) from low-left to up-right.
This show the positioning of the component THK.

General case

a2 SET : a2 THK with number of LEAD = s = 2m + 1

a2 may be an empty set = 0

They **start on one of the left side (mandrel) boundaries and go on to right side corresponding boundaries**

LEFT **RIGHT**

1……………………………………………………………………a2

2……………………………………………………………………a2 - 1

3……………………………………………………………………a2 - 2

a2 - 1…………………………………………………………….2

a2…………………………………………………………………1

a1 SET : a1 THK with number of LEAD = s = 2m - 1

LEFT **RIGHT**

a2 + 1…………………………………………………………….A

a2 + 2……………………………………………………………. A - 1

a2 + 3……………………………………………………………. A - 2

A - 1……………………………………………………………. a2 + 2

A…………………………………………………………………. a2 + 1

When a2 = 0 then the S-H-P-K is made of **a1 = A interlocked component THK** with **2m – 1 LEAD**

Observe on the image just after this point the LEFT RIGHT boundaries couples

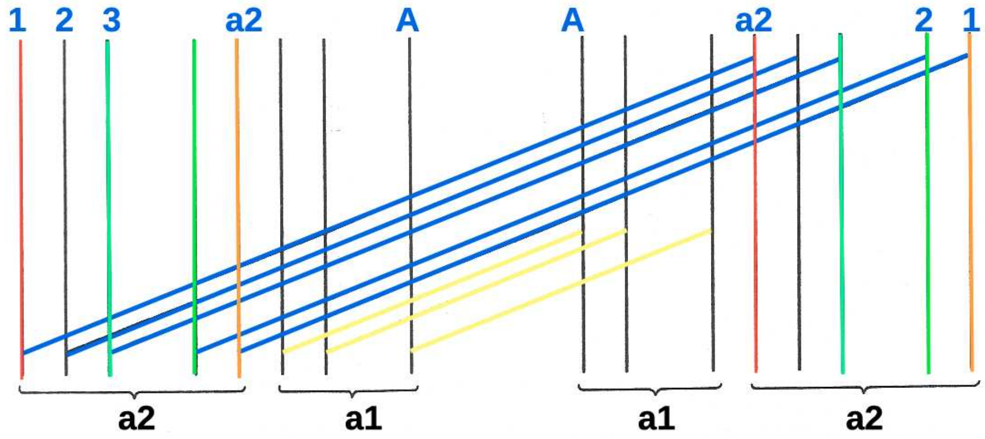

Fig. 29 — The set of a_1 half-cycles with $s = 2m - 1$, and the set of a_2 half-cycles with $s = 2m + 1$.

REWRITTEN from SCHAAKE & TURNER

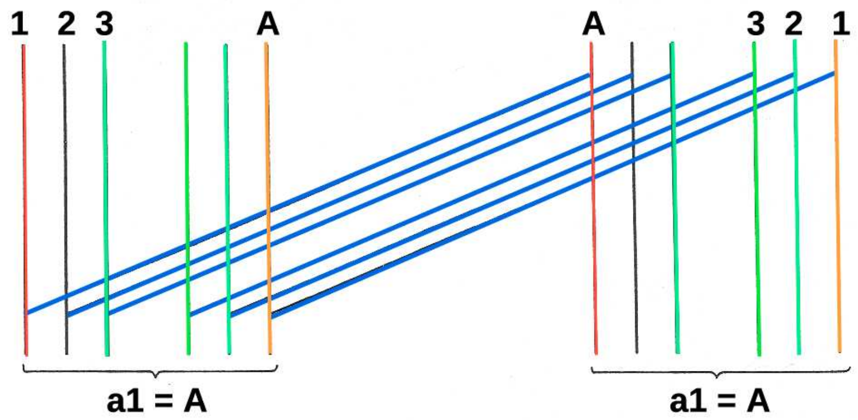

Fig. 30 — $a_2 = 0$ and hence $a_1 = A$ with $s = 2m - 1$.