



## MATER - A simple Mate Searching Program

(c) Valentin Albillo, 2020



Last update: 14/01/98

### Description:

**MATER** is a simple mate searching program, implemented as an MS-DOS command. Given a legal position in FEN notation, the side to move, and the maximum number of moves to give mate, it will search and output either a move which gives mate in that number of moves or less, or the fact that there's no such move.

### Main features:

- Full legal move generation, including full legal castling, *en passant* pawn captures and underpromotions.
- Can find mates in any number of moves, up to 64, subject to available memory and time.
- Can find general mates or mates in which the mating side only gives checks. This is useful for finding much quicker long sequences of checks that end in mate.
- Accepts positions in FEN notation, and checks them for legality and syntax.
- Searches for mates recursively, and thus always finds the shortest possible mate.
- The search can be halted immediately at any moment, simply pressing any key.
- Outputs the first move that gives the required mate, or the fact that one doesn't exist.
- Shows progress for each iteration, including time taken.
- Outputs final total time, as well as nodes examined.
- DOS command format: takes its parameters from the command line, and upon termination sets `ERRORLEVEL` with a code indicating the outcome of the search.
- Can be invoked from any program which can execute DOS commands (i.e: Visual Basic).
- The output can be redirected to a file, the printer, another program, etc.
- If executed without parameters, displays syntax and an example.
- Fully commented, easy to understand, no-tricks, didactic source code for you to study and use in your own chess programs if you want to.
- Absolutely free.

### What it isn't:

- Being a didactic example, it isn't optimized nor does it use the most efficient but highly complex algorithms there are for board representation and move generation. The emphasis has been placed on **didactic, simple-to-understand** code.
- As a result, its running speed is nothing to write home about, and can't generally compete with more elaborate code. In other words, it can be **slow**.

However, some simple heuristics have been used to speed the search and, in the case of searching for mates consisting of a sequence of checks, it can find them much faster than a more general approach. For instance, in the last example featured it finds a **mate in 12** in just **0.40 seconds**. See examples below.

### Syntax:

**MATER** [FEN position] [max.mov(1..64)] [all/checks (a/c)]

where:

- **FEN position** is the position on the board, including the side to move, in FEN notation.

Example: The initial position would be: `rnbnqkbnr/pppppppp8/8/8/8/PPPPPPPP/RNBQKBNR/ w`

- **max.mov(1..64)** is the maximum number of moves to give mate, from 1 to 64.
- **all/checks (a/c)** specifies the type of mate to search for:
  - **a:** general mates, *all legal moves* will be tried.
  - **c:** *only checking moves* for the mating side will be tried.

## Download:

### IMPORTANT NOTE:

I'VE THOROUGHLY CHECKED THE EXECUTABLE TO MAKE SURE IT'S VIRUS FREE, BUT I ACCEPT NO RESPONSIBILITIES. IF YOU DOWNLOAD THE EXECUTABLE, CHECK IT OUT WITH YOUR OWN ANTIVIRUS SOFTWARE BEFORE USING IT AT ALL. ALTERNATIVELY, YOU COULD COMPILE THE SOURCE CODE.

Please note that apart from making freely available the code in this Section, and heavily commenting both source code and executable (including examples where appropriate), I cannot give any further support, specially via e-mail, so please don't ask any questions about this software nor advice about how to integrate it with your own programs, and such. I simply do not have the time. Thanks for understanding.

### REQUIRED LEGALESE (If you're normal people, give it a miss):

All the code in this section has been written by myself specifically for it, I am the copyright holder, but you can download and use it, either in executable form and/or source code, in all or in part, as you wish, including its use as part of your own programs, be they freeware or commercial, absolutely free. The only thing you should not do is claim you wrote my routines. Apart from that, use them as you wish. Credit would be appreciated, but not required. Of course, no warranty of any kind is given at all. NO SUPPORT OF ANY KIND IS OFFERED. I don't claim the code is suitable or adequate for any purpose whatsoever, nor that it's bug free. I shall not be held responsible of any damage their use can do. You use it at your own risk. Your using the code indicates acceptance of these terms. If you are not willing to accept them, do not use the code.

You can download either the **MS-DOS executable command**, suitable for Intel PC processors (8088, 8086, 286, 386, 486, Pentium, etc) or any good MS-DOS emulator (Linux **DOSEMU**, for instance), and/or the **Turbo Pascal** source code.

- To **view** the source code, simply click this link with the *left* mouse button. To **download** the source code to your hard disk, click the link with the *right* mouse button and chose **Save Link as ...** in the menu that appears:

### **MATER Turbo Pascal Source code (42 Kb)**

**Note:** The Turbo Pascal source code can be compiled with version 7.0 of Turbo Pascal (or Borland Pascal) or higher, and can be adapted to Borland Delphi and other Pascal implementations (or even MS Visual Basic) quite easily.

- To **download** the MS-DOS executable to your hard disk, simply click the following link with the *left* mouse button:

### **MATER executable MS-DOS command (30 Kb)**

**IMPORTANT NOTE:** This **won't work** if you are accessing this page as a PDF document because links have been removed from the current (2020) PDF versions of the original HTML pages.

## Documentation:

**MATER** has been implemented as an MS-DOS command. You execute it at the DOS prompt either in real MS-DOS or in a DOS shell from within Windows or Linux **DOSEMU**. Memory requirements are absolutely minimal. See the proper **Syntax** above.

If executed without parameters, or with wrong or illegal parameters, **MATER** terminates immediately after printing a line describing what went wrong and a brief resume of proper syntax, including an example, like this:

### **MATER**

```
INPUT ERROR: wrong number of parameters
```

```
Syntax: MATER [FEN position] [max.mov (1..64)] [all/checks (a/c)]  
Example: mater b7/PP6/8/8/7K/6B1/6N1/4R1bk/ w 5 a
```

```
Note: The output can be redirected: MATER ... > mat.txt  
Exit: - mate in n found: ERRORLEVEL = n  
      - mate not found: ERRORLEVEL = 0  
      - halted by user: ERRORLEVEL = 9999
```

- **MATER** accepts positions in **FEN notation**, including board and side to move. The FEN position should be in this form:  
(row1)/(row2)/(row3)/.../(row8)/ (side to move, b or w)

The following is checked and has to be true for legality:

- Only characters 'p','n','b','r','q','k','P','N','B','R','Q','K' may be used to represent Black and White pieces.  
Example: 3S4/2C5/ ... is illegal, S and C aren't valid.
  - Exactly 8 rows should be specified.  
Example: b7/PP6/8/8/7K/6B1/6N1/ w is illegal, only 7 rows.
  - In each row, exactly 8 squares must be specified, either with pieces or an adequate number representing blank squares in the row.  
Example: b7/PP6/8/8/7K/6B1/6N/4R1bk/ w is illegal, only 7 squares.
  - Each side can only have a maximum of 8 pawns, a maximum of 16 pieces, and exactly one King.  
Example: b7/PP6/8/8/7k/6B1/6N1/4R1bk/ w is illegal, no White King.
  - The last row should end with a slash (/) and then the side to move (b or w) must be specified, separated by at least one space.  
Example: b7/PP6/8/8/7K/6B1/6N1/4R1bk/w is illegal, no space before the 'w'.
- Apart from the FEN position (including side to move) you must specify a **maximum** number of moves in which to search mate (from 1 to 64), and if a general mate should be found (**all**), or a special mate in which the mating side **only makes checks** (checks).
  - If the syntax is incorrect, you will get *one* of these messages:

```
INPUT ERROR: max.mov should be an integer
INPUT ERROR: max.mov should be between 1 and 64
INPUT ERROR: Must specify A (all Moves) or C (checks only)
INPUT ERROR: Must specify who moves: B (Black) or W (White)
INPUT ERROR: Illegal FEN position
```

**Note:** The routine that performs the parsing, checking, and conversion from FEN notation to a board position is called **FEN2Posit**. It accepts a string representing the FEN position, fills an internal structure with the corresponding board position, and returns a boolean indicating if it was successful or not. See the source listing.

This routine automatically assigns castling rights depending on whether King and Rooks are on their original positions or not. Although the board structure does permit it, no provision has been taken to allow the user to specify castling rights and/or the square for a possible initial *en passant* capture.

- Once the syntax is correct, **MATER** prints its name and version, a copyright notice, the text representation of the board position (together with castling rights and *en passant* square, if any), the position in FEN notation, the maximum number of moves to mate, and the type of mate (all moves or only checks) being sought, like this:

```
MATER: Mate searching program v1.1. (c) Valentin Albillo

      a b c d e f g h
# # # # # # # # # # White can't castle long
8 # b - - - - - - - # White can't castle short
7 # P P - - - - - - # Black can't castle long
6 # - - - - - - - # Black can't castle short
5 # - - - - - - - # No en passant square
4 # - - - - - - K #
3 # - - - - - B - #
2 # - - - - - N - #
1 # - - - - R - b k #
# # # # # # # # # #

FEN: b7/PP6/8/8/7K/6B1/6N1/4R1bk/ w

Max. no. of moves to mate: 3
Searching *all moves* for: White
```

- Then, an iterative search is made, beginning with a mate in 1, till finding a mate or reaching the maximum number of moves. This ensures the **shortest** possible mate is found.

For each root move searched, a dot (.) is printed, and after all moves have been searched the time taken for that iteration is printed and the next one begins:

```
Mate in 1.....0.01
Mate in 2.....0.18
Mate in 3.....
```

**Note:** You can immediately stop the search at any moment by simply pressing any key. **MATER** will terminate at once after printing a message specifying it was halted by the user.

- Once a mate has been found, the search stops immediately and **MATER** terminates at once after printing the mating move and some information, such as total time used and number of nodes visited:

```
Mate in 3 with b7xa8=N, t=0.10, nodes=1978
```

**Note:** The output of **MATER** can be **redirected** to the printer, a file, or as the input of another command. This is done as with all other DOS commands. Consult your MS-DOS manual or helpfile for details on redirection.

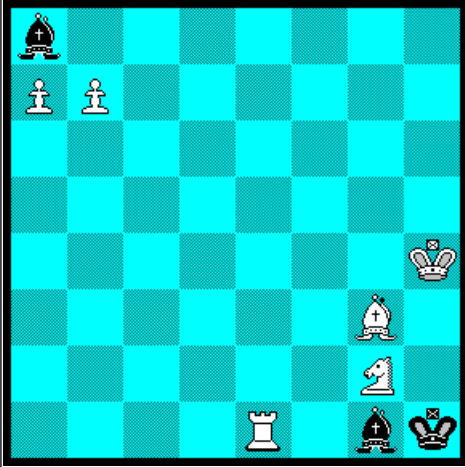
After **MATE** terminates for whatever reason, the MS-DOS environment variable **ERRORLEVEL** is set to a value indicating the cause for termination. Possible values are:

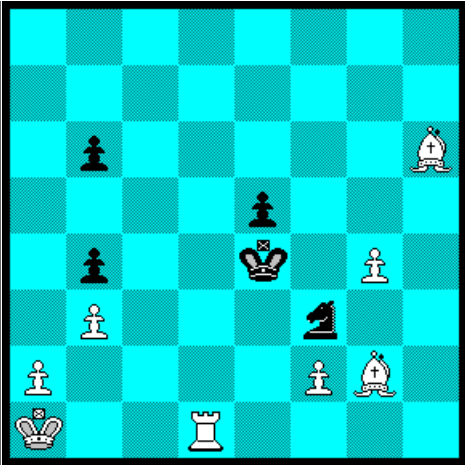
- ERRORLEVEL = n indicates a **mate in n** was found
- ERRORLEVEL = 0 indicates **no mate** was found
- ERRORLEVEL = 9999 indicates the user **halted MATER**

If desired, this value can be checked from inside a batch (.BAT) file to take proper actions. Consult your MS-DOS documentation for details on writing and using BAT files.

- Now, some examples of **MATER** in action, with challenging mates including rare **underpromotions**, long sequences of checks, etc.

### MATER examples (run on the HP Vectra/Pentium MMX/200 Mhz)

	<pre>MATER b7/PP6/8/8/7K/6B1/6N1/4R1bk/ w 3 a MATER: Mate searching program v1.1. (c) Valentin Albillo    a b c d e f g h # # # # # # # # # # White can't castle long 8 # b - - - - - - - # White can't castle short 7 # P P - - - - - - # Black can't castle long 6 # - - - - - - - - # Black can't castle short 5 # - - - - - - - - # No en passant square 4 # - - - - - - - K # 3 # - - - - - - B - # 2 # - - - - - - N - # 1 # - - - - R - b k # # # # # # # # # # #  FEN: b7/PP6/8/8/7K/6B1/6N1/4R1bk/ w Max. no. of moves to mate: 3 Searching *all moves* for: White  Mate in 1.....0.00 Mate in 2.....0.00 Mate in 3..... Mate in 3 with b7xa8=N, t=0.10, nodes=1978</pre>
<p>If you want the full solution and other interesting data on this mate problem, see:</p> <p style="text-align: center;"><b>Amazing Problem 01</b></p>	



If you want the full solution and other interesting data on this mate problem, see:

**Amazing Problem 02**

MATER 8/8/1p5B/4p3/1p2k1P1/1P3n2/P4PB1/K2R4/ w 3 a

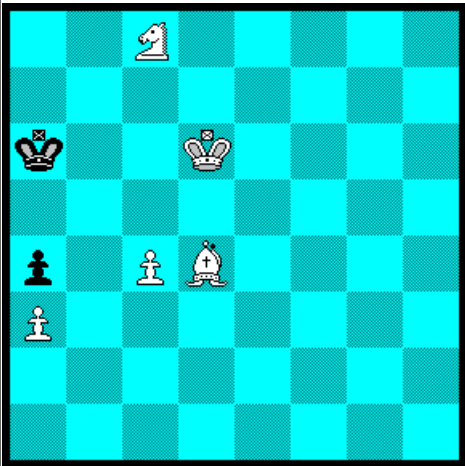
MATER: Mate searching program v1.1. (c) Valentin Albillo

	a	b	c	d	e	f	g	h	
#	#	#	#	#	#	#	#	#	White can't castle long
8	#	-	-	-	-	-	-	-	White can't castle short
7	#	-	-	-	-	-	-	-	Black can't castle long
6	#	-	p	-	-	-	-	B	Black can't castle short
5	#	-	-	-	p	-	-	-	No en passant square
4	#	-	p	-	-	k	-	P	#
3	#	-	P	-	-	-	n	-	#
2	#	P	-	-	-	-	P	B	-
1	#	K	-	-	R	-	-	-	#
#	#	#	#	#	#	#	#	#	#

FEN: 8/8/1p5B/4p3/1p2k1P1/1P3n2/P4PB1/K2R4/ w

Max. no. of moves to mate: 3  
Searching \*all moves\* for: White

Mate in 1.....0.00  
Mate in 2.....0.00  
Mate in 3.....  
Mate in 3 with Bh6-c1, t=0.06, nodes=2178



If you want the full solution and other interesting data on this mate problem, see:

**Amazing Problem 03**

MATER 2N5/8/k2K4/8/p1PB4/P7/8/8/ w 4 a

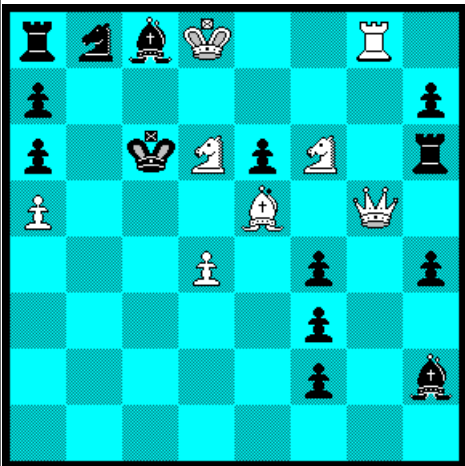
MATER: Mate searching program v1.1. (c) Valentin Albillo

	a	b	c	d	e	f	g	h	
#	#	#	#	#	#	#	#	#	White can't castle long
8	#	-	-	N	-	-	-	-	White can't castle short
7	#	-	-	-	-	-	-	-	Black can't castle long
6	#	k	-	-	K	-	-	-	Black can't castle short
5	#	-	-	-	-	-	-	-	No en passant square
4	#	p	-	P	B	-	-	-	#
3	#	P	-	-	-	-	-	-	#
2	#	-	-	-	-	-	-	-	#
1	#	-	-	-	-	-	-	-	#
#	#	#	#	#	#	#	#	#	#

FEN: 2N5/8/k2K4/8/p1PB4/P7/8/8/ w

Max. no. of moves to mate: 4  
Searching \*all moves\* for: White

Mate in 1.....0.00  
Mate in 2.....0.00  
Mate in 3.....0.11  
Mate in 4....  
Mate in 4 with Kd6-c7, t=0.57, nodes=15315



If you want the full solution and other interesting data on this mate problem, see:

**Amazing Problem 06**

MATER rnbK2R1/p6p/p1kNpN1r/P3B1Q1/3P1p1p/5p2/5p1b/8/ w 4 a

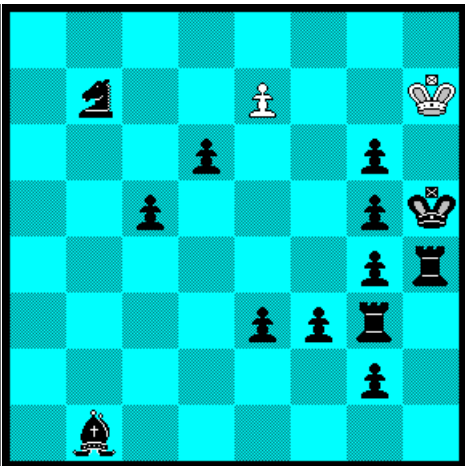
MATER: Mate searching program v1.1. (c) Valentin Albillo

	a	b	c	d	e	f	g	h	
#	#	#	#	#	#	#	#	#	White can't castle long
8	#	r	n	b	K	-	-	R	#
7	#	p	-	-	-	-	-	p	#
6	#	p	-	k	N	p	N	-	r
5	#	P	-	-	-	B	-	Q	-
4	#	-	-	-	P	-	p	-	p
3	#	-	-	-	-	p	-	-	#
2	#	-	-	-	-	p	-	b	#
1	#	-	-	-	-	-	-	-	#
#	#	#	#	#	#	#	#	#	#

FEN: rnbK2R1/p6p/p1kNpN1r/P3B1Q1/3P1p1p/5p2/5p1b/8/ w

Max. no. of moves to mate: 4  
Searching \*all moves\* for: White

Mate in 1.....0.00  
Mate in 2.....0.04  
Mate in 3.....2.50  
Mate in 4.....  
Mate in 4 with Qg5-g2, t=67.01, nodes=1203866



If you want the full solution and other interesting data on this mate problem, see:

### Amazing Problem 07

MATER 8/1n2P2K/3p2p1/2p3pk/6pr/4ppr1/6p1/1b6/ w 3 a

MATER: Mate searching program v1.1. (c) Valentin Albillo

```

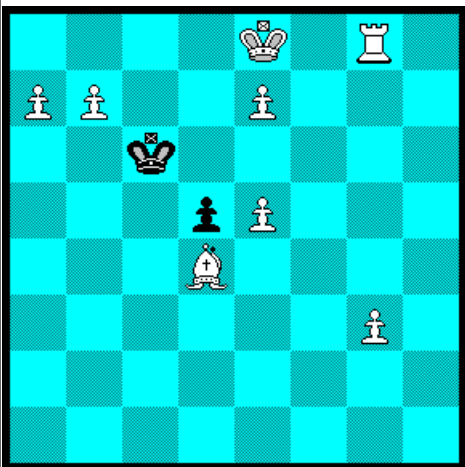
a b c d e f g h
# # # # # # # # # # White can't castle long
8 # - - - - - - - # White can't castle short
7 # - n - - - P - - K # Black can't castle long
6 # - - - - p - - p - # Black can't castle short
5 # - - p - - - - p k # No en passant square
4 # - - - - - - - p r #
3 # - - - - - p p r - #
2 # - - - - - - - p - #
1 # - b - - - - - - #
# # # # # # # # # #

```

FEN: 8/1n2P2K/3p2p1/2p3pk/6pr/4ppr1/6p1/1b6/ w

Max. no. of moves to mate: 3  
Searching \*all moves\* for: White

Mate in 1.....0.00  
Mate in 2.....0.00  
Mate in 3.....  
Mate in 3 with Kh7-g7, t=0.15, nodes=1170



If you want the full solution and other interesting data on this mate problem, see:

### Amazing Problem 09

MATER 4K1R1/PP2P3/2k5/3pP3/3B4/6P1/8/8/ w 3 a

MATER: Mate searching program v1.1. (c) Valentin Albillo

```

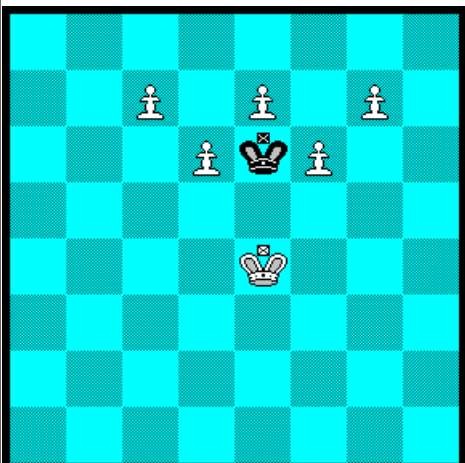
a b c d e f g h
# # # # # # # # # # White can't castle long
8 # - - - - - K - R - # White can't castle short
7 # P P - - - P - - - # Black can't castle long
6 # - - k - - - - - # Black can't castle short
5 # - - - - p P - - - # No en passant square
4 # - - - - - B - - - - #
3 # - - - - - - - P - #
2 # - - - - - - - - #
1 # - - - - - - - - #
# # # # # # # # # #

```

FEN: 4K1R1/PP2P3/2k5/3pP3/3B4/6P1/8/8/ w

Max. no. of moves to mate: 3  
Searching \*all moves\* for: White

Mate in 1.....0.00  
Mate in 2.....0.00  
Mate in 3.....  
Mate in 3 with b7-b8=R, t=0.20, nodes=4805



Here **underpromotions** are involved ...

MATER 8/2P1P1P1/3PkP2/8/4K3/8/8/8/ w 3 a

MATER: Mate searching program v1.1. (c) Valentin Albillo

```

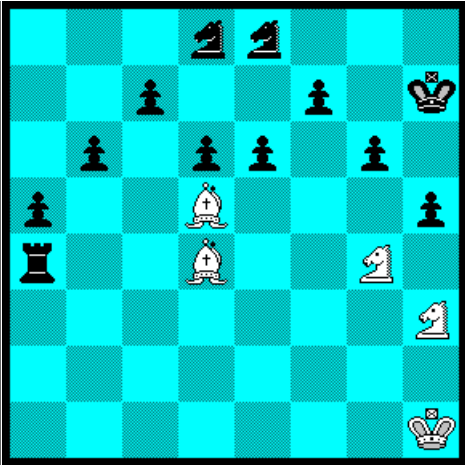
a b c d e f g h
# # # # # # # # # # White can't castle long
8 # - - - - - - - - # White can't castle short
7 # - - P - - P - P - # Black can't castle long
6 # - - - - P k P - - # Black can't castle short
5 # - - - - - - - - # No en passant square
4 # - - - - - K - - - #
3 # - - - - - - - - #
2 # - - - - - - - - #
1 # - - - - - - - - #
# # # # # # # # # #

```

FEN: 8/2P1P1P1/3PkP2/8/4K3/8/8/8/ w

Max. no. of moves to mate: 3  
Searching \*all moves\* for: White

Mate in 1.....0.00  
Mate in 2.....0.00  
Mate in 3.....  
Mate in 3 with e7-e8=B, t=0.13, nodes=2288



Here the king is checkmated after a long sequence of checks ...

MATER 3nn3/2p2p1k/1p1pp1p1/p2B3p/r2B2N1/7N/8/7K/ w 12 c

MATER: Mate searching program v1.1. (c) Valentin Albillo

	a	b	c	d	e	f	g	h			
#	#	#	#	#	#	#	#	#	#	White can't castle long	
8	#	-	-	-	n	n	-	-	#	White can't castle short	
7	#	-	-	p	-	-	p	-	k	#	Black can't castle long
6	#	-	p	-	p	p	-	p	-	#	Black can't castle short
5	#	p	-	-	B	-	-	-	p	#	No en passant square
4	#	r	-	-	B	-	-	N	-	#	
3	#	-	-	-	-	-	-	-	N	#	
2	#	-	-	-	-	-	-	-	-	#	
1	#	-	-	-	-	-	-	-	K	#	
#	#	#	#	#	#	#	#	#	#	#	

FEN: 3nn3/2p2p1k/1p1pp1p1/p2B3p/r2B2N1/7N/8/7K/ w

Max. no. of moves to mate: 12  
 Searching only checks for: White

Mate in 1.....	0.00
Mate in 2.....	0.00
Mate in 3.....	0.00
Mate in 4.....	0.00
Mate in 5.....	0.00
Mate in 6.....	0.01
Mate in 7.....	0.03
Mate in 8.....	0.06
Mate in 9.....	0.11
Mate in 10.....	0.20
Mate in 11.....	0.38
Mate in 12.....	
Mate in 12 with Nh3-g5, t=0.40, nodes=9342	