

## Stacks of Coins: Test Cases

I created several test cases with a simple tool `createtest [S] [TargetHeight] [NumberMoves]`. The parameters determine the number of stacks, the target height of the stacks after flattening and the number of moves the test case generator should do (this is not the number of moves you have to do at a minimum).

I wrote two programs `stacks1.cpp` and `stacks2.cpp`. The latter (mentioned as "Advanced" in the table) implements the idea of pruning redundant permutations. The times are taken on a Pentium 3 with 667 MHz. Compiler optimization was turned off.

Test case	Stacks	Start	Moves needed	Simple	Advanced
1	8	8 1 2 14 12 9 12 6	6	5.0s	0.4s
2	2	51 13	5	0.0s	0.0s
3	6	28 36 8 4 12 8	5	0.1s	0.0s
4	6	21 33 9 3 24 54	6	0.5s	0.1s
5	6	28 40 4 4 16 4	6	0.1s	0.0s
6	3	15 77 100	7	0.0s	0.0s
7	4	14 1 69 44	7	0.1s	0.1s
8	3	8 87 1	7	0.0s	0.0s
9	6	28 43 3 4 16 2	8	6.0s	0.7s
10	7	9 11 6 5 6 9 10	8	—	0.4s
11	6	56 86 6 8 32 4	8	6.0s	0.7s
12	4	7 1 69 51	8	0.2s	0.1s

The contestants output can differ from the sample output but it should use the same number (or less ;-)) moves.