Generalised Hanoi Disks

Andy Pepperdine

We all probably know of the "Hanoi discs" or "Tower of Hanoï" [2], introduced to the West by E Lucas in 1883. There are three pegs and a number of discs piled on one of them with the largest on the bottom, with progressively smaller ones above. These are to be moved to one of the other pegs, but only one disc can be moved at a time, and no disk can be placed on a smaller one. How many moves are necessary to complete the move?

One generalisation is to increase the number of pegs. Clearly it will take fewer moves if four pegs are available, and fewer still as the number of pegs are increased. H.E. Dudeney [3] cast this as the "Reve's Puzzle".

Commentary

The solution to the 3-peg problem is well-known; it needs 2^n-1 disc moves, where n is the number of discs in the initial pile, and is easily found. Hint: move the smallest one every other move.

When more pegs are added, the number of moves needed shrinks dramatically, but it is not easy to find a formula or algorithm that can be followed to carry out the task. It can be described recursively, easily, but that is not much use when presented with the situation in physical reality!

Define the function $f(p,d)$ to be a function that returns the number of moves to make for p pegs and d discs. Then imagine an intermediate state during the solution where there is a pile of discs on the initial peg, and another pile on one other peg (even if this only immediately after the first disc is moved). Then the solution consists of creating this intermediate state, moving the remainder from the initial peg to the target peg, and then repeating the first step (in reverse order) to move the intermediate pile onto the target. So we can define the number of moves required for p pegs and d discs:

$$f(p,d) = \min_{1<=k<d} \{2.f(p,k) + f(p-1,d-k)\}$$

with special cases for $f(3,d)$ and $f(p, 1)$, which are left to the reader.

There are a number of optimisations that can be performed when writing a computer program for this problem which are left as an exercise for the reader.

Solution

The following table shows the results for lower numbers of pegs and discs.

# discs	Number of pegs							
	3	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	1	1	1	1	1	1	1	1
2	3	3	3	3	3	3	3	3
3	7	5	5	5	5	5	5	5
4	15	9	7	7	7	7	7	7
5	31	13	11	9	9	9	9	9
6	63	17	15	13	11	11	11	11
7	127	25	19	17	15	13	13	13
8	255	33	23	21	19	17	15	15
9	511	41	27	25	23	21	19	17
10	1023	49	31	29	27	25	23	21
11	2047	65	39	33	31	29	27	25
12	4095	81	47	37	35	33	31	29
13	8191	97	55	41	39	37	35	33
14	16383	113	63	45	43	41	39	37
15	32767	129	71	49	47	45	43	41
16	65535	161	79	57	51	49	47	45
17	131071	193	87	65	55	53	51	49
18	262143	225	95	73	59	57	55	53
19	524287	257	103	81	63	61	59	57
20	1048575	289	111	89	67	65	63	61
21	2097151	321	127	97	71	69	67	65
22	4194303	385	143	105	79	73	71	69
23	8388607	449	159	113	87	77	75	73

# discs	Number of pegs							
	3	**4**	**5**	**6**	**7**	**8**	**9**	**10**
24	16777215	513	175	121	95	81	79	77
25	33554431	577	191	129	103	85	83	81
26	67108863	641	207	137	111	89	87	85
27	134217727	705	223	145	119	93	91	89
28	268435455	769	239	153	127	97	95	93
29	536870911	897	255	161	135	105	99	97
30	1073741823	1025	271	169	143	113	103	101

References

[1] Creative Commons License, *http://creativecommons.org/licenses/by/2.5/*

[2] W.W Rouse Ball (ed. HSM Coxeter), *Mathematical Recreations and Essays*, Macmillan, 1967

[3] H.E. Dudeney, *The Canterbury Puzzles*, reprinted by Dover, 1958

[4] J.T. Boardman, C. Garrett, G.C.A. Robson, "A Recursive Algorithm for the Optimal Solution of a Complex Allocation Problem using a Dynamic Programming formulation", *The Computer Journal, Vol 29, no. 2, p182,* 1986.

[5] J.S. Rohl, T.D.Gedeon, "The Reve's Puzzle", *The Computer Journal, Vol 29, no. 2, p187,* 1986.

2007-07-01 (format edited 2010-01-02)