Towers of Hanoi, revisited

Kees van der Laan

Abstract

Another version of TEXing of "The Towers of Hanoi" problem is provided, which does not assume Lisp knowledge, just plain TEX. Also some variations of use are included, among others to remove the restriction on the disks: disk size can be supplied by more than one digit.

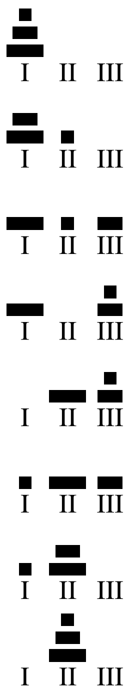
1 Introduction

Along with the Dedham TUG91 conference, I attended David Salomon's advanced TEX course. Despite his clear and ample exercises I decided to simplify Leban (1985).

2 The Towers of Hanoi problem

A pyramid of disks —means a tower with implicit ordering of the disks— has to be replaced under the restrictions that only one disk at a time can be moved, each intermediate state consists of pyramids —obeying the original implicit ordening. In total three pyramids can be used. For a pyramid of n disks, the solution needs $2^n - 1$ moves. For an introduction to the problem see the opening paragraphs in Graham c.s. (1989).

3 Example



is obtained by

```
\newcount\n \n=3
\def\I{123}
%Input auxiliaries and macros
\Hanoi\I\II\III\n
```

4 Macros

I chose to represent the pyramid via defs and to supply the number of disks in a counter.

```
\def\Hanoi#1#2#3#4{%Moves from #1 to #2,
%with aid of pyramid #3.
%The number of disks is #4.
\ifnum#4=1 %For Tower of 1 disk,
           %just move the disk
   \movedisk from#1to#2%
   \showtowers%Print towers after move
\else%Problem of #4 disks is solved by
     %problem of (#4-1) disks, a move and
     %again a problem of (#4-1) disks.
   {\advance#4 by-1 \Hanoi#1#3#2#4}%
   \movedisk from#1to#2%
   \showtowers%Print towers after move
   {\advance#4 by-1 \Hanoi#3#2#1#4}%
\fi}%end \Hanoi
%
\def\movedisk from#1to#2{%Move disk from
%tower #1 to tower #2
    \def\move##1##2\emove{%Separate first
    %disk from #1, move that, and restore
    %stub of #1 in #1
       \pre#2by##1%Precede tower #2 by top
                  %of tower #1
       \gdef#1{##2}%Restore stub #1
                          }%end \move
\ea\move#1\emove%Move from #1 to #2
}%end \movedisk
```

For printing,

```
\def\showtowers{%Display towers
\hbox{\vertib\I\ \vertib\II\ %
      \vertib\III}\par}
%
%Vertical printing(Based on Salomon's notes)
\def\gobble#1{}%To eat character
\newbox\ver %Will contain typesetted tower
\newdimen\diskhigh %Height of each disk
```

```
 \diskhigh=1ex\relax%
\def\vertib#1{%Typeset tower #1
 \setbox\ver=\vbox{}%Will contain tower
 \ea\vertclb#1\end\endd%\ver contains tower
 \vbox to\hgt ex{\vfil\unvbox\ver\kern.5ex%
%              To separate identification
 \hbox to\brd ex{\hss\ea\gobble\string#1%
\hss}}%Tower and identification, typesetted
}%end\vertib
\def\vertclb#1#2\endd{%
  \ifx\end#1%Empty tower
  \else%Put disk #1 in auxiliary hbox
   \setbox0=\hbox to\brd ex{%
   \hss\vrule width#1ex%Size disk #1
          height\diskhigh\relax\hss}%
     %Next add disk to \ver
   \setbox\ver=\vbox{%
    \unvbox\ver\kern.5ex\box0}%Add new disk
   \vertclb#2\end\endd%Add rest of disks
  \fi}%end \vertclb
```

Auxiliaries

```
%To be initiated: \n and \I
\newcount\n  \n=3  %The number of disks
\def\I{123}        %The start tower
%Program proper
\nopagenumbers
\def\II{} \def\III{}
\newcount\brd \brd=\n %Breadth of towers
\newcount\hgt %Height of maximum tower
\hgt=\n \multiply\hgt by2 \advance\hgt by1
\let\ea=\expandafter
\def\pre#1by#2{%Precede string #1 (stored
%in a command) by the character #2.
   \ea\gdef\ea#1\ea{\ea#2#1}%
}%end \pre
```

5 Variations

5.1 Only the number of disks as argument

It would be nice and look simpler to provide only the number of disks as argument, for example via

```
 \hanoi\n
```

The consequence is that behind the screens the pyramid `\I` has to be defined as 1, 2, ... n (≤ 10), and that the old `\Hanoi\I\II\III\n` has to be invoked. The only difficulty is to create `\I`. This can be done in a loop and by the use of `\pre` along with redefinition of `\I`.

5.2 Disks not restricted to one digit

This looks like a serious problem but it is not. The basic approach is to provide commands for numbers consisting of more than one digit. For example one could consider the pyramid 10, 11, 12.

How to do this?

In principle via

```
\n=3    %Number of disks
\def\x{10}\def\xi{11}\def\xii{12}
\brd=12 %Breadth of largest disk
\def\I{\x\xi\xii}
\Hanoi\I\II\III\n
```

5.3 Generalized disks

Up till now we have considered disks consisting essentially out of numbers. What about for example (xyz) as disk, provided via a command? The only problem is the printing because we don't know the size of it. So let us assume for printing that just the contents of the pyramids will do in the implicit provided order. This can be done via the following simplified `\showtowers`.

```
\def\showtowers{\vskip1ex
  \ \hbox to 3ex{\ea\gobble\string\I:
                \hss}\I\par
  \ \hbox to 3ex{\ea\gobble\string\II:
                \hss}\II\par
  \ \hbox to 3ex{\ea\gobble\string\III:
                \hss}\III\par}
```

Example

A tower consisting of 1, ab, (xyz) can be moved via[1]

```
\n=3
\brd=5
\def\ab{ab}\def\xyz{(xyz)}
\def\I{1\ab\xyz}
```

where for printing use is made of the above provided `\showtowers`.

5.4 Interactively?

Downes (1991) examplified how to use `\message` for communicating with the user. One could think of a modification of `\showtowers` with appropriate `\message` commands, such that the moves will appear on the screen. For oneliners the `\immediate\write16{...}` can better be used I could see that it worked; no previewing nor printing. The modified `\showtowers` reads

```
\def\showtowers{
\message{\ea\gobble\string\I:   \I}
\message{\ea\gobble\string\II:  \II}
\message{\ea\gobble\string\III: \III}}
```

6 Conclusion

While still in progress it seemed appropriate to me to release a proof version for NTG's Fall91 Meeting 'Fun with TEX' at Eindhoven.

References

[1] Downes, M.J.(1991): Dialogue with TEX. Proceedings TUG91.

[2] Graham, R.L, D.E. Knuth, O. Pastashnik (1989): Concrete Mathematics. Addison-Wesley.

[3] Leban, B.(1985): A solution to the Tower of Hanoi problem using TEX. *TUGboat* 6(3), 151–154.

[1] The ab and xyz defs can take a space before and after the replacement text. In print the disks will be separated then.