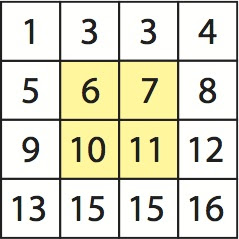
**A** purely (?) algorithmic [Le Monde mathematical puzzle](https://xianblog.wordpress.com/2011/09/03/le-monde-puzzle-website/)

*For the table below, what is the minimal number of steps required to reach equal entries when each step consists in adding ones to three entries sitting in a* L*, such as (7,11,12) or (5,6,10)? Same question for the inner table of four in yellow.[](http://www.affairedelogique.com/espace_probleme.php?corps=probleme&num=1069)*

For the inner table, this is straightforward as there are four possible L’s, three equations like 6+n⁶=7+n⁷,  and two degrees of freedom leading to a unique entry of N=13 (impossible!)  or 16 (feasible). Hence adding 10 L’s. For the entire table, summing up all entries after completion leads to 16N, which is also equal to 1+3+3+…+16+M, where M is the number of added L’s, itself equal to 138+3O, if O denotes the number of ones added. Hence M can only take the values 18, 21, … It took me quite a while to R code an approach to complete the table into 16 18’s, as my versions of simulated annealing did not seem to converge. In the end, I used a simplified version where the table was completed by multinomial draws, like a M(17;3⁻¹,3⁻¹,3⁻¹) for the upper left corner, corresponding to random draws of one of the 36 available L’s, which should be used 50 times in total, and then augmented or reduced of one L depending on the value at a randomly selected entry. Leading to the result

> aneal(grid=c(1,3,3:13,15,15,16),maxT=1e5)

[1] 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18

The R code is quite simple-minded if a wee bit long, using a preliminary definition of the 36 L’s as a 26×3 matrix named allels:

aneal=function(gri,horz=1e3){

numbels=rep(0,36)

sumz=rep(0,16)

while (sum(numbels)<50){ #50 L's used in toto

i=sample(1:16,1)

if (sum(numbels[apply(i==allels,1,max)==1])<18-gri[i]){

rez=18-gri[i]-sum(numbels[apply(i==allels,1,max)==1])#leftover

indz=(1:36)[apply(i==allels,1,max)==1]

indz=indz[numbels[indz]==0]#empty L's

indz=sample(rep(indz,2),rez,rep=TRUE)

for (j in indz) numbels[j]=numbels[j]+1

}}

t=1

while (t18){#remove one L

subz=indz[numbels[indz]>0]#used local L's

j=sample(rep(subz,2),1)

numbels[j]=numbels[j]-1}

if (sumz<18){#add one local L

j=sample(rep(indz,2),1)

numbels[j]=numbels[j]+1}

for (i in 1:16)#check constraints

sumz[i]=sum(numbels[apply(i==allels,1,max)==1])+gri[i]

if ((min(sumz)==18)&(max(sumz)==18)) break()

t=t+1}

print(sumz)}