A Visual Proof of Amortised-linearResizable Arrays

Don Blaheta
Knox College
Galesburg, IL, USA
dblaheta@knox.edu

ABSTRACT
We demonstrate visually why doubling capacity is the better
strategy when resizing arrays. The visual proof makes sim-
ple amortised analysis more accessible to a CS2 audience.

Categories and Subject Descriptors
K.3.2 [Computers and Education]: Computer and Infor-
mation Science Education—Computer science education

General Terms
Algorithms

Keywords
Amortised analysis, visual proof

1. THE PROOF
CS2 students are usually quick to understand that increas-
ing an array’s size by only one, necessitating a resize with
every single addition, is too expensive. Their usual first
suggestion, however, is to increase the size by a constant
amount. We can diagram the amount of work required for
each strategy as follows (here with an interval of three):

We draw the amount of copying work at time \( i \) as a
width-1 vertical bar whose height corresponds to the number
of operations. On the left, the total amount of work can be
computed geometrically: the area of the triangle with base
\( n \) and height \( n \) is \( \frac{n^2}{2} \). The geometric analysis of the diagram
on the right is slightly less obvious.

To make it clearer, we perform a diagram transformation
that makes each vertical bar “tip over” or “melt” into the
open space to its left, creating a wider, shorter rectangle
equal in area to the original:

We now have a jagged-edged triangle with an area, or total
copying work load, of (about) \( \frac{1}{2} \cdot n \cdot \frac{3}{2} \) or \( \frac{n^2}{6} \).

The stage is now set for a visual proof of the amortised-
linear cost of resizing at increasing intervals. We now start
with a size of three and double whenever full; and now when
we “tip over” the rectangles into the spaces to their left, all
but the first fall to a height of exactly 2:

The work done fits strictly within a rectangle of area \( 2n \).

2. CONCLUSIONS
Visual proofs are not new, and the fifth diagram above is
similar to one in at least one CS2 textbook.[1] However, we
believe that the “tip over” diagram transformation is new,
and the visual proof is a compelling one for CS2 students
who have never seen amortised analysis before. The trans-
formation is exactly analogous to the well-known accounting
technique of saving two credits or “cyber-dollars” per add
operation for the eventual copy-and-resize.

3. REFERENCES
[1] M. T. Goodrich and R. Tamassia. Data structures and
algorithms in Java. 2006.