# Practice with loopy question 3 and 4 using array diagrams

When we work with array diagrams as assertions, making progress toward termination in the repetend usually means making some segment smaller.

Our first example involves moving the even elements in an array to the beginning, by swapping array elements. Here is the information needed to write the repetend:

!B && P   implies   R

P: b

0	h	j	b.length
even	?	odd	

B: h ≤ j
Decrease: the size of b[h..j]
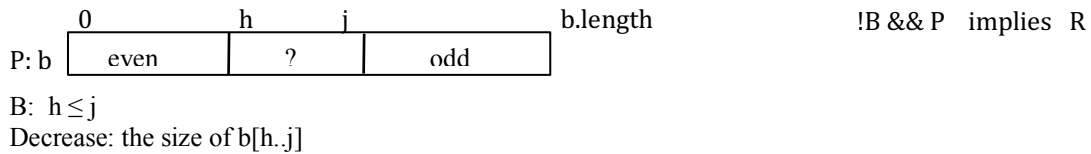
You have seen something like this before, so let us try something different here. We try to make as few array elements swaps as possible. Looking at segment b[h..j], we can see three cases:

1. b[h] is even. In this case, increase h, for that decreases the size of b[h..j] while keeping the invariant true.
2. b[j] is odd. In this case, decrease j, for that decreases the size of b[h..j] while keeping the invariant true.
3. b[h] is odd and b[j] is even. Swap b[h] and b[j], putting an even value in b[h] and an odd value in b[j]. Then increase h and decrease j.

This gives this repetend:

> **if** (b[h] is even) h= h+1;
> **else if** (b[j] is odd) j= j–1;
> **else** {Swap b[h] and b[j]; h= h+1; j= j–1;}

Earlier, we developed a repetend for a similar problem by looking only at b[h]. We can also write a repetend by looking only at b[j]. Here are the two repetends:

> **if** (b[h] is even) h= h+1;
> **else** {Swap b[h] and b[j]; j= j–1;}

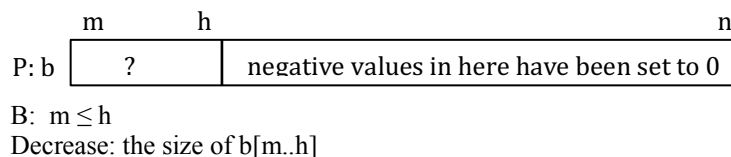> **if** (b[j] is odd) j= j–1;
> **else** {Swap b[h] and b[j]; h= h+1;}

These two solutions use only one test per iteration, but each swap may put only one value in its final place. The repetend we developed above may make up to two tests per iteration, but each swap puts two values in their final place.

Do you see how using loop invariants helps us easily develop and compare different algorithms for the same problem? Without loop invariants, you would not be able to see or discuss such possibilities so easily.

Exercise

Consider a loop to set all negative values of array segment b[m..n] to 0. Below is the information you need to write the repetend —decreasing the given expression while keeping the invariant true. Write the repetend. The answer appears at the end of the script for this video.

This example comes from an algorithm to search a sorted array b for a value x. Precondition Q and invariant P are given below. This example is similar enough to the previous one that you should be able to it yourself. Please do that. The answer is at the end of the pdf of this script.

P: b

m	h		n
?	negative values in here have been set to 0		

B: m ≤ h
Decrease: the size of b[m..h]

Answer to exercise

if (b[h] < 0) b[h]= 0;
h= h–1;