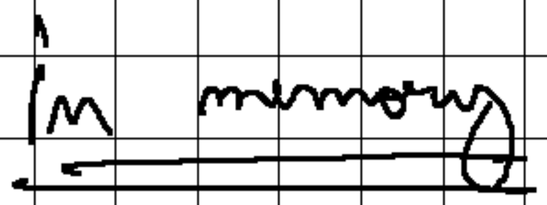
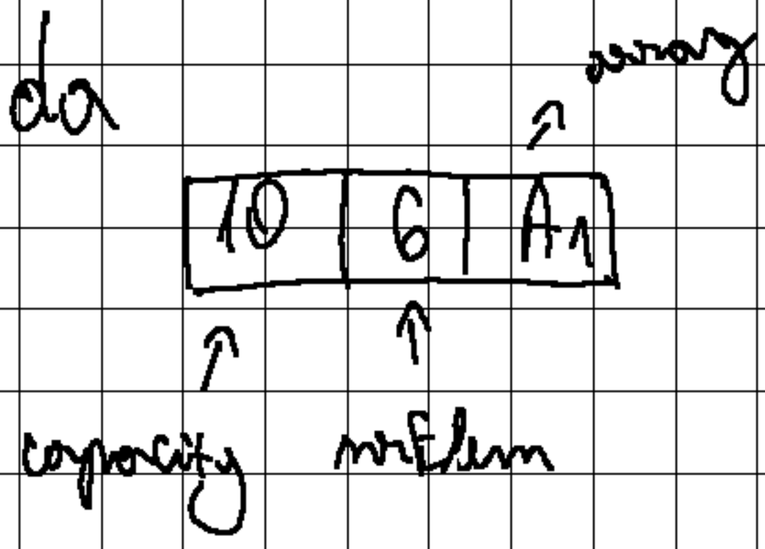
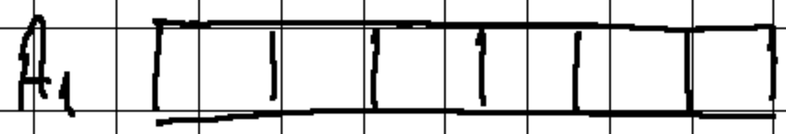
Subalgorithm addToEnd (da, elem) is
    if da.cap == da.nrElems then
        newElems ← @ a new array with da.cap*2
        for i ← 1, da.nrElems execute
            newElems[i] ← da.elems[i]
        endfor
        da.cap ← da.cap*2
        @ maybe deallocate the old array
        da.elems ← newElems
    endif
    // adding to a dynamic array
    da.elems[da.nrElems+1] ← elem
    da.nrElems ← da.nrElems+1
end addToEnd

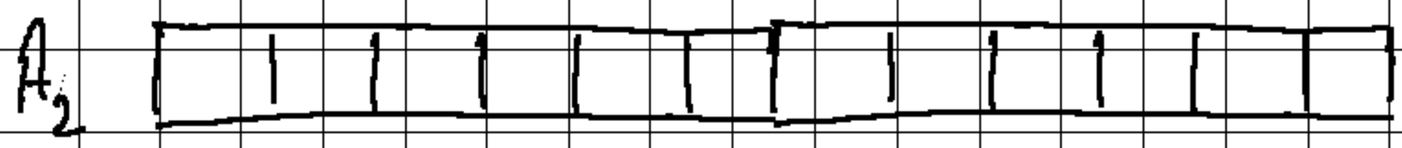
Complexity: BC: $\Theta(1)$
WC: $\Theta(nrElems)$
TC: $O(nrElems)$

```
subalg addToPosition(da, elem, i) is
    if i<1 or i>da.nrElems+1 then
        @ throw an exception
    endif
    // resizing a dynamic array
    if da.cap == da.nrElems then
        newElems ← @ a new array with da.cap*2
        for i←1, da.nrElems execute
            newElems[i] ← da.elems[i]
        endfor
        da.cap ← da.cap*2
        @ maybe deallocate the old array
        da.elems ← newElems
    endif
    for j=da.nrElem, i, -1 execute
        da.elems[j+1] ← da.elems[j]
    endfor
    da.elems[i] ← elem
    da.nrElem ← da.nrElem+1
```


m
memory


da
array
10
6
A1
capacity
nrElem


A1


A2