Expandable Array

- An expandable array is a data structure that changes size, as needed, as elements are inserted
- To be expandable, an array must have been dynamically-allocated
- There is usually no limit on the size of such arrays, other than the size of the main memory (heap)
- How we "expand" an array:
  1. When an array is full, dynamically create a larger array
  2. Copy over the values from the old array to the new array
  3. Assign the new array to the existing array variable (pointer)
  4. Delete the old array using delete [ ]

NOTE:

- Also called *dynamic array*, *growable array*, *resizable array*, *dynamic table*, *mutable array*

Expandable Array

elementCount = 4

oldArray

6	1	7	8		
0	1	2	3	4	5

Expandable Array

elementCount = 4

Insert element at oldArray[elementCount]

insert 5

oldArray

0	1	2	3	4	5
6	1	7	8		

Expandable Array

elementCount = 5

oldArray

6	1	7	8	5	
0	1	2	3	4	5

Expandable Array

elementCount = 5

Insert element at oldArray[elementCount]

insert 2

oldArray

6	1	7	8	5	
0	1	2	3	4	5

Expandable Array

elementCount = 6

insert 3

oldArray

6	1	7	8	5	2
0	1	2	3	4	5

Expandable Array

elementCount = 6

Insert element at newArray[elementCount]

insert 3

oldArray

6	1	7	8	5	2
0	1	2	3	4	5

The array is full and there is no room for a new element!

Expandable Array

elementCount = 6

insert 3

oldArray

6	1	7	8	5	2
0	1	2	3	4	5

So we will create a new larger array …

Expandable Array

elementCount = 6

insert 3

oldArray

6	1	7	8	5	2
0	1	2	3	4	5

So we will create a new larger array …

Here it is:

newArray

0	1	2	3	4	5	6	7	8	9	10	11

Expandable Array

elementCount = 6

insert 3

oldArray

6	1	7	8	5	2
0	1	2	3	4	5

… and copy the elements of the old array into it …

newArray

0	1	2	3	4	5	6	7	8	9	10	11

Expandable Array

elementCount = 6

insert 3

oldArray

6	1	7	8	5	2
0	1	2	3	4	5

… and finally insert 3 into the new array.

Insert element at newArray[elementCount]

newArray

6	1	7	8	5	2						
0	1	2	3	4	5	6	7	8	9	10	11

Expandable Array

elementCount = 7

oldArray

6	1	7	8	5	2
0	1	2	3	4	5

We now need to delete [ ] oldArray

newArray

6	1	7	8	5	2	3					
0	1	2	3	4	5	6	7	8	9	10	11

Expandable Array

elementCount = 7

newArray

6	1	7	8	5	2	3					
0	1	2	3	4	5	6	7	8	9	10	11

Expandable Array – Size?

- How large should our new array be?
  - When answering this question, we need to consider the following facts:
    - Every time we expand an array, we need to copy its elements into the new array
      -> time consuming
      - So we do not want to do the expansion often
    - Right after the expansion, a portion of the array is empty
      -> not space efficient
      - So we do not want to expand the array by too much

Expandable Array – Size? (cont'd)

- Possible answers:
  - Expanding the array by 1 cell?
    - But expanding after each insertion will be very slow because we need to copy the elements from the old array into the new array

      -> time consuming: O(n)
  - Expanding the array by doubling the number of cells?
    - This works well because the array grows very large very quickly: 10, 20, 40, 80, 160, 320, 640, 1280, …
    - Therefore, very few array expansions ar required
    - And the cost of all insertions is amortized to O(1)

For more information, please, see https://en.wikipedia.org/wiki/Dynamic_array