Lecture Topics:

- C Basics
- Dynamic Array

Abstract Data Type (ADT)

- Abstract Data Type (ADT) – a mathematical model for data types
- Specifies:
  - the type of data stored
  - the operations supported on them
  - the types of parameters of the operations.

- Why “abstract”?
  - an implementation-independent view of the data type

Dynamic Arrays

- Elements in an array are stored in a contiguous block of memory
- Allow random access (direct access)
  - i.e., time to access the 1st element = time to access the last element
  - By using array subscript ([]):

```
int* array = malloc(1000 * sizeof(int));
array[0] = 0;
array[999] = 0;
```

size - 1

- Demo…

Dynamic Arrays (cont.)

- Basic operations:
  - get – Gets the value of the element stored at a given index in the array
  - set – Sets/updates the value of the element stored at a given index in the array
  - insert – Inserts a new value into the array at a given index.
    - Sometimes, dynamic array implementations limit insertion to a specific location in the array, e.g. only at the end.
  - remove – Removes an element at a given index from the array
    - Sometimes, dynamic array implementations avoid moving elements up a spot by only allowing the last element to be removed

Dynamic Arrays (cont.)

- Drawbacks:
  - Fixed size, must be specified when the array is created
    - For static array:

```
int array[50];
```

    - For dynamic array:

```
int *array = malloc (50 * sizeof(int));
```

→Need to allocate more memory if we need to store more data

  - How?

- Dynamic array DS doesn’t have a fixed capacity
  - Has a variable size and can grow as needed

Dynamic Arrays (cont.)

- Need to keep track of three things:
  - data – underlying data storage array
  - size – number of elements currently stored in the array
  - capacity – number of elements data has space for before it must be resized
- How it works?
  - An array of known capacity is maintained by the dynamic array DS.
  - As elements are inserted, they are simply stored in data
  - If an element is inserted into the dynamic array, and there isn't capacity for it in the underlying data storage array (data), the capacity of the underlying data storage array is doubled. Then the new element is inserted into this larger data storage array.

Dynamic Arrays

5	

5	8

5	8	1	

5	8	1	4

5	8	1	4	9			

5	8	1	4	9	0		

5	8	1	4	9	0	6	

5	8	1	4	9	0	6	7

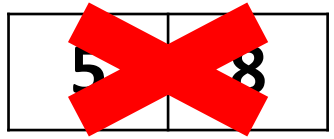
5	8	4	9	0	6	7	

5	8	4	9	6	7		

Inserting an element into dynarray

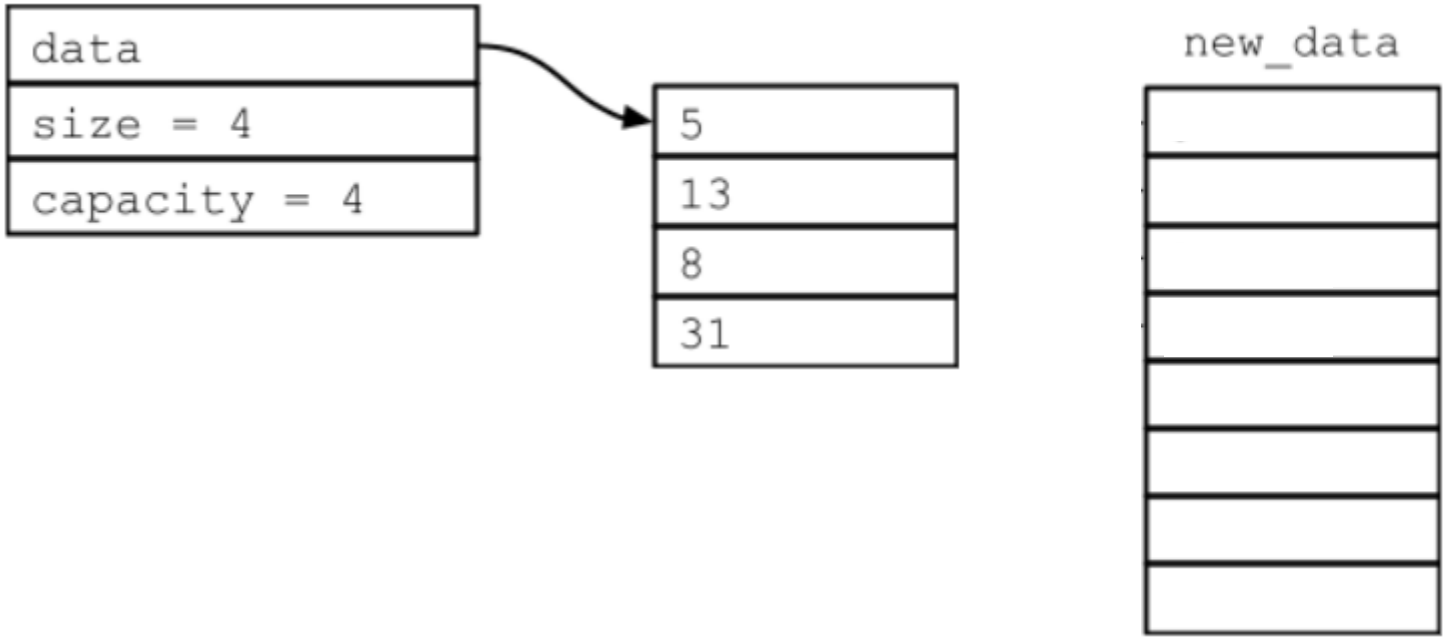
- Case 1: if size < capacity
  - At least one free spot in data
  - Insert the new element

- Case 2: if size == capacity
  - No free spot in data
  - Step 1: allocate a new array that has twice the capacity
  - Step 2: copy all elements from data to new array
  - Step 3: delete the old data array
  - Step 4: Insert the new element



Another Example

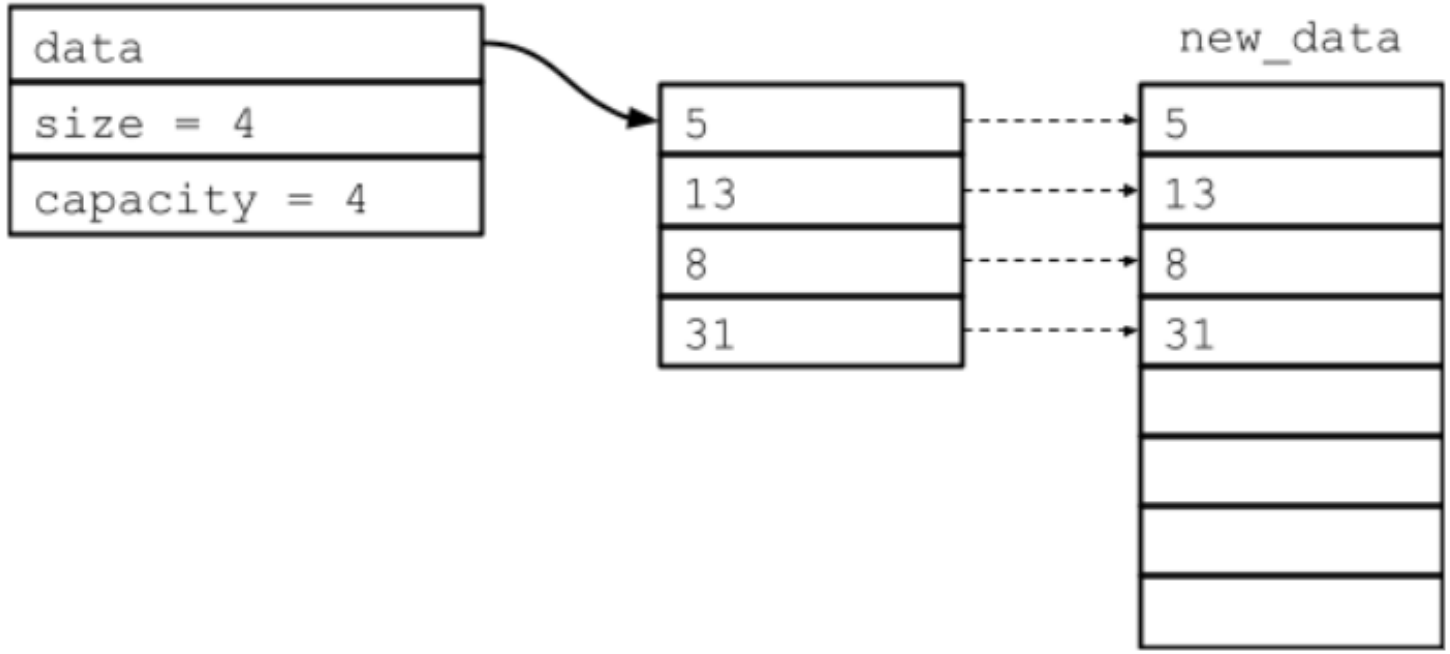
- Insert 16 to the following dynamic array:



- Step 1: allocate a new array that has twice the capacity

Another Example

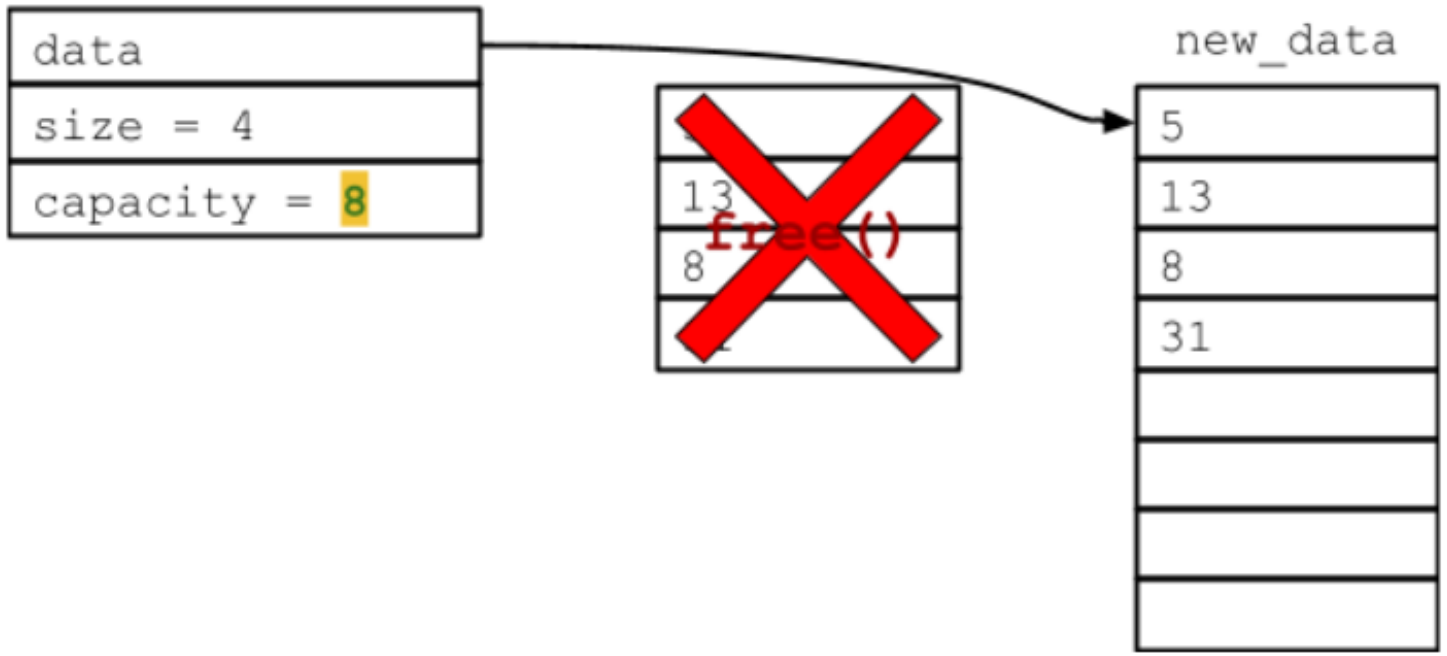
- Insert 16 to the following dynamic array:



- Step 2: copy all elements from data to new array

Another Example

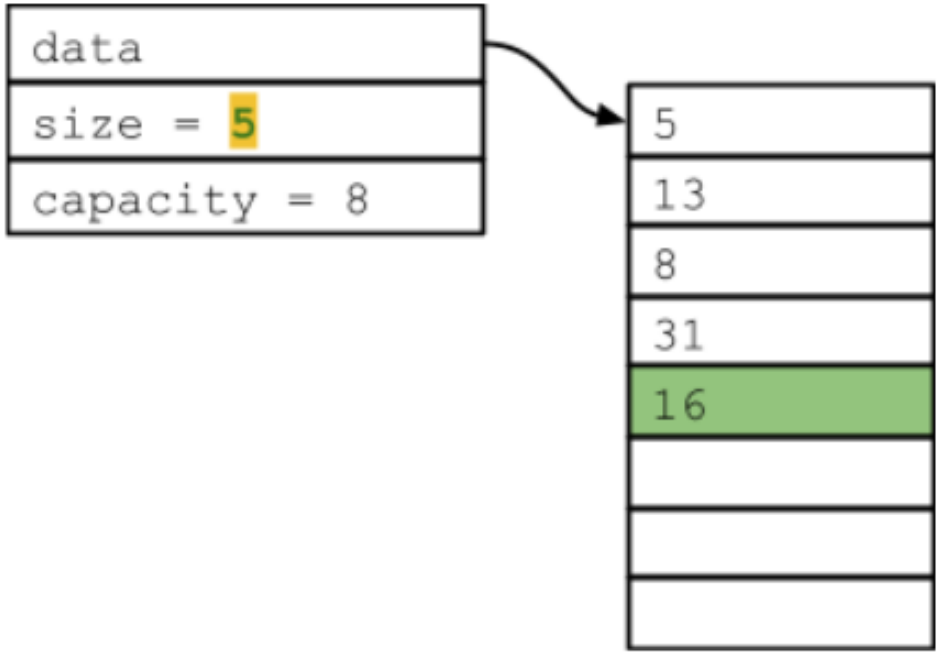
- Insert 16 to the following dynamic array:



- Step 3: delete the old data array and update data

Another Example

- Insert 16 to the following dynamic array:



- Step 4: Insert the new element