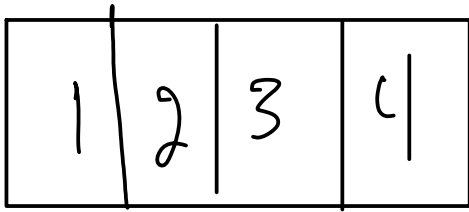
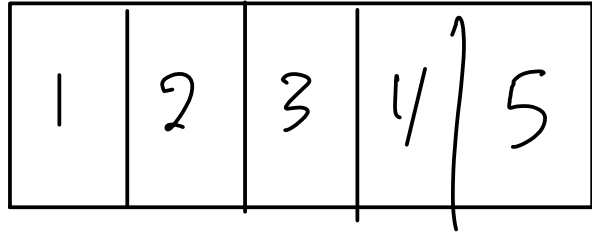


Idea: When out of space, copy to an array 1 larger.



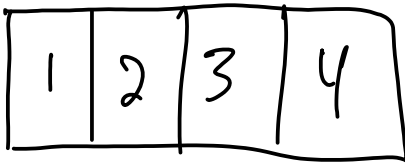
$O(n)$



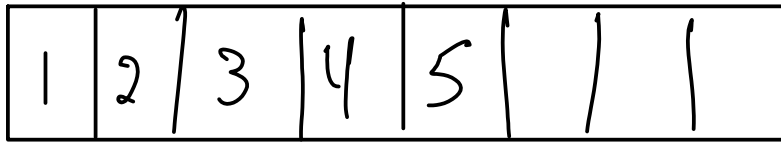
For  $n$  insertions:  
 $O(n^2)$

| ins | ops |
|-----|-----|
| 1   | 1   |
| 2   | 2   |
| 3   | 3   |
| 4   | 4   |
| 5   | 5   |

Idea: when full, double



Start: empty, size 4



$O(n)$

| ins | ops |
|-----|-----|
| 1   | 1   |
| 2   | 1   |
| 3   | 1   |
| 4   | 1   |
| 5   | 5   |
| 6   | 1   |
| 7   | 1   |
| 8   | 1   |
| 9   | 9   |

6  $O(1)$

7  $O(1)$

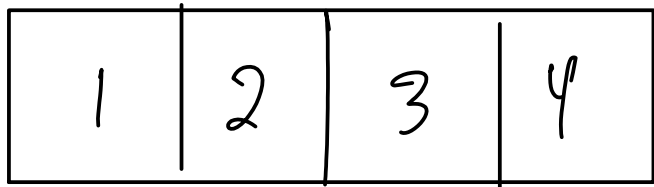
8  $O(1)$

|   | budget | spend | left over<br>cumulative |         |
|---|--------|-------|-------------------------|---------|
| 1 | 2      | 1     | 1                       |         |
| 2 | ↓      | 1     | 2                       |         |
| 3 |        | 1     | 3                       |         |
| 4 |        | 1     | 4                       |         |
| 5 |        | 5     | $6 - 5 = 1$             |         |
| 6 |        | 1     | 2                       |         |
| 7 |        | 1     | 3                       |         |
| 8 |        | 1     | 4                       |         |
| 9 |        |       | 9                       | $6 - 9$ |

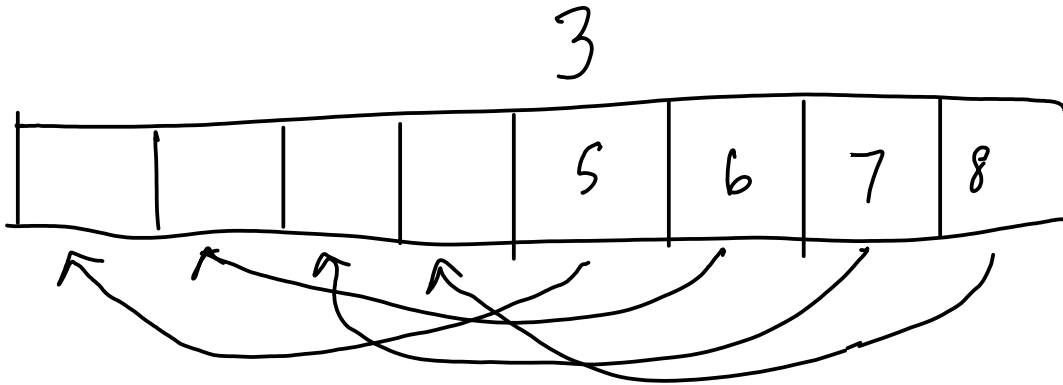
Initially empty,  
size of 4

~~$6 - 9$~~  oops

|   | allocate | spend | bank           |                  |
|---|----------|-------|----------------|------------------|
| 1 | 3        | 1     | 2              |                  |
| 2 | ↓        | 1     | 4              |                  |
| 3 |          | 1     | 6              |                  |
| 4 |          | 1     | 8              |                  |
| 5 |          | 5     | $11 - 5 = 6$ ← |                  |
| 6 |          | 1     | 8              |                  |
| 7 |          | 1     | 10             |                  |
| 8 |          | 1     | 12             |                  |
| 9 |          |       | 9              | $15 - 9 = 6$ ← ★ |



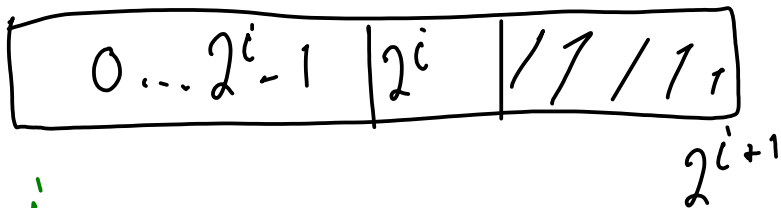
- 3 tokens pay for:
- initial insertion
  - moving yourself
  - moving somebody in the first half of the array



Immediately after resize

from  $2^i$  to  $2^{i+1}$

Assume 0 left over tokens  
 $2^{i+1} - 1$



$(2^{i+1} - 1) - 2^i$  ins  
 $2^i - 1$  ins  $\rightarrow$  full

$2^i - 1$  ins  $\rightarrow$  2 left over tokens each  $2(2^i - 1)$

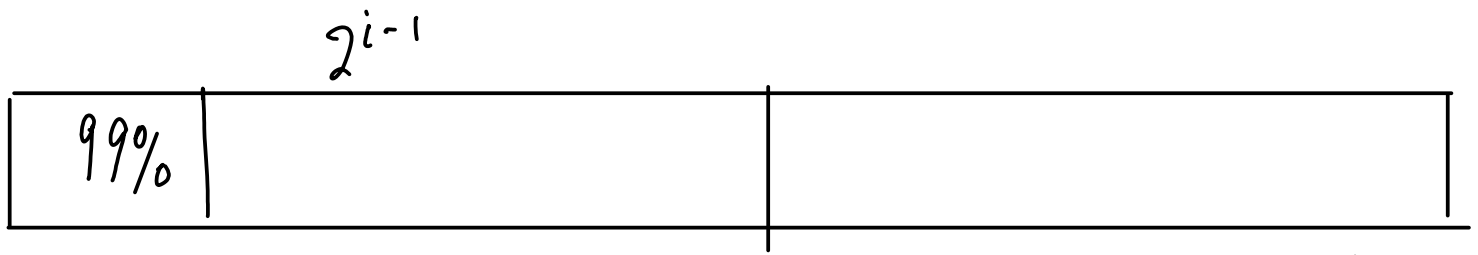
$2^i$  get (3) tokens have  $2^{i+1} + 1$  tokens to spend  $= 2^{i+1} - 2$

$2^{i+1}$  copies  $\rightarrow$   $2^{i+2}$  array  
 1 extra ins

Need to spend  $2^{i+1} + 1$  tokens.

We spend exactly as many tokens as we have.

When to shrink? How about halve array when half full,



Problem sequence  $O(n^2)$  alternate ins/del on either side of a resize.  $2^{i+1}$

Instead:

When 1/4 full, cut in half

$n$  : ins + del  $\rightarrow O(n)$

3 tokens for ins, 2 for del

Bit counter

K-bits

1 op  $\rightarrow$  flip 1 bit

0000  $\rightarrow$  0001 1 flip  $\leftarrow$

0011  $\rightarrow$  0100 3 flips

if LSB = 0  $\rightarrow$  1 flip