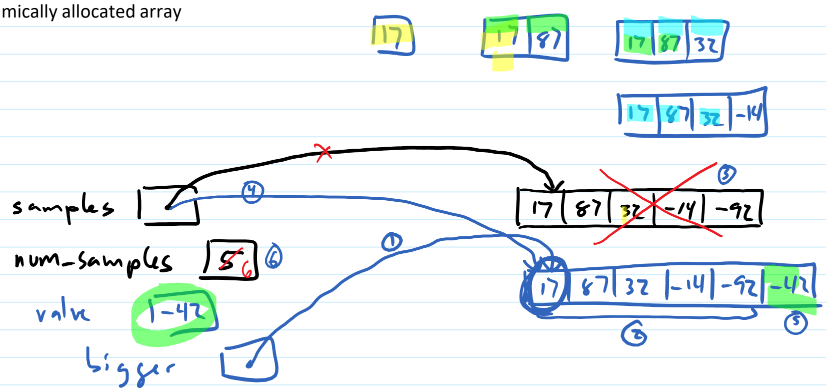
Resizing a dynamically allocated array

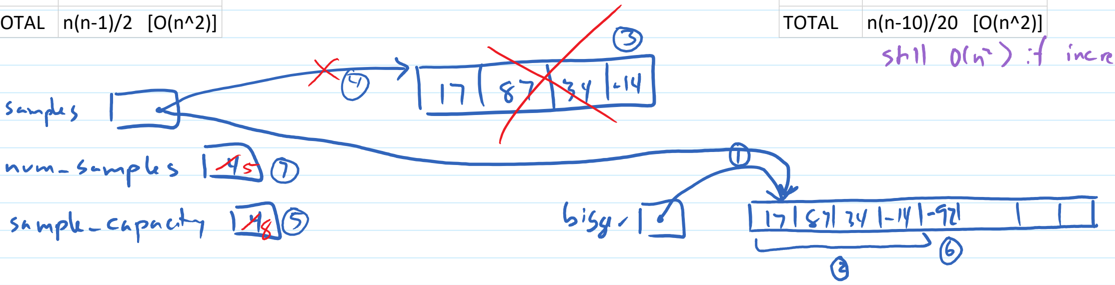


```
   int *samples = NULL;
   int num_samples = 0;

   while (scanf("%d", &value) > 0)
     {
     ① int *bigger = malloc(sizeof(int) * (num_samples + 1));

       for (int i = 0; i < num_samples; i++)
   ②   {
         bigger[i] = samples[i];
       }

     ③ free(samples)
     ④ samples = bigger;

     ⑤ samples[num_samples] = value;
     ⑥ num_samples++;
       ...
     }
}
```

resize	number of copies
1 -> 2	1
2-> 3	2
3 -> 4	3
4-> 5	4
...	...
n-1 -> n	n-1
TOTAL	n(n-1)/2 [O(n^2)]

resize	number of copies
10 -> 20	10
20 -> 30	20
30 -> 40	30
40-> 50	40
...	...
n-10 -> n	n-10
TOTAL	n(n-10)/20 [O(n^2)]

still $O(n^2)$ if increasing size by 10 each resize



```
    if (num_samples == samples_capacity)
       {
       ①int *new_samples = malloc(sizeof(int) * samples_capacity * 2);

       ②memcpy(new_samples, samples, sizeof(int) * num_samples);

       ③free(samples);

       ④samples = new_samples;
       ⑤samples_capacity *= 2;
       }

   ⑥samples[num_samples] = value;
   ⑦num_samples++;
```

resize	number of copies
1 -> 2	1
2-> 4	2
4 -> 8	4
8-> 16	8
...	...
n/2 -> n	n/2
TOTAL	n-1 [O(n)]

(assuming n is a power of 2)

still O(n) if increasing by 1%, 10%, 50%, 200%, ...