CSCI 136
Data Structures &
Advanced Programming

Conference

(Vector Growth)

Vectors: Add Method Complexity

Suppose we grow the Vector's array by a fixed amount `d`.

How long does it take to add `n` items to an empty Vector?

- The array will be copied each time its capacity needs to exceed a multiple of `d`
  - At sizes `0d, 1d, 2d, … , (n/d)d.`
- Copying an array of size `k*d` takes `c*k*d` steps for some constant `c`, giving a total of:

$$\sum_{k=1}^{n/d} ckd \;=\; cd\,\sum_{k=1}^{n/d} k \;=\; cd\,\left(\frac{n}{d}\right)\left(\frac{n}{d}+1\right)/2 \;=\; O(n^2)$$

Vectors: Add Method Complexity

Suppose we instead grow the Vector's array by doubling.

How long does it take to add `n` items to an empty Vector?

- The array will be copied each time its capacity needs to exceed a power of 2
  - At sizes `0, 1, 2, 4, 8 …,` $2^{\lfloor \log_2 n \rfloor}$
    - Because the final array copy will occur at size $\lfloor \log_2 n \rfloor$
- The total number of elements are copied when `n` elements are added is:

$$1 + 2 + 4 + \cdots + 2^{\lfloor \log_2 n \rfloor} = \sum_{i=0}^{\lfloor \log_2 n \rfloor} 2^i$$

Induction to the Rescue!

- In the induction video, we proved that

$$\sum_{i=0}^{k} 2^i = 2^{k+1} - 1$$

- So

$$\sum_{i=0}^{\lfloor \log_2 n \rfloor} 2^i = 2^{(\lfloor \log_2 n \rfloor)+1} - 1 \le 2n - 1 = O(n)$$

- Thus, the average amount of overhead due to copying the array is a constant amount per element.
- Thus, the Vector add method runs in *amortized constant time!*