CS330HW8

ras70

November 2017

Problem 2

Aggregate method

Consider the process P - taking the bit string of zeros and calling the increase operation until the bit string is again all zeros. This process takes 2^n increase operations. The largest number that can be represented by a bit string of length n is $2^n - 1$ (where all of the bits are one). Therefore, it takes $2^n - 1$ increase operations to go from 0 to the largest number possible, and an additional increase call to get from the largest number possible back to 0, resulting in 2^n total calls.

Let t_i be the cost of the i^{th} increase operation. Let T_p be the total running time of the process P, increasing the bit string from 0 all the way back to 0. Since this process takes 2^n increase operation to complete,

$$T_p = \sum_{i=1}^{2^n} t_i$$

Let b_i be the total cost of flipping bit i throughout the entire process P. In other words, b_i is the accumulated cost of bit i in 2^n increase operations. The total running time of the process P is equal to the sum of b_i for all n bits. More formally,

$$T_p = \sum_{i=1}^{n} b_i$$

Observe, that in 2^n increase operations the k^{th} bit is flipped $\frac{2^n}{2^{k-1}}$ times. The first bit is flipped on every operations (2^n times), the second bit is flipped every other time ($2^n/2$ times), the third bit is flipped every 4 times ($2^n/4$ times)..etc. The cost of flipping a bit through the n increase operations is the cost of flipping the bit once, times the number of times it is flipped. Since the cost per flip of bit k is given as 2^k, the total cost of flipped bit i through n increase operations is,

$$b_i = \sum_{j=1}^{\frac{2^n}{2^{k-1}}} 2^k = \frac{2^n * 2^k}{2^{k-1}} = 2 * 2^n = 2^{n+1}$$

Therefore, the cost of flipping each bit k over 2^n increase operations is equal to 2^{n+1}. Plugging this into the expression for T_p yields:

$$T_p = \sum_{i=1}^{n} 2^{n+1} = n * 2^{n+1}$$

This means the total running time of 2^n increase operations is equal to $n * 2^{n+1}$. The amortized running time A_t is equal to the total running time of x operations divided by x - where x is chosen to be a given number of operations that helps signify the average running time. More formally, in this case, $A_t = T_p/2^n$. Therefore, $A_t = \frac{n*2^{n+1}}{2^n} = 2n$. Since the amortized running time is equal to $2n$, it must be true that the amortized running time is bounded by n, or

$$A = O(n)$$