CS 261 - Data Structure **Spring 2017**

HomeWork 1

April 14, 2017 *Liangjian Chen*

Problem 1 **Solution:**

When received a ”startag”, push the name of tag into the stack.When received an ”endtag”, check whether the top of stack is same as the incoming tag name. If it is same, pop the top of stack, otherwise report an error occur.

Problem 2 **Solution:**

My algorithm is maintaining two array. One for enqueue another one for dequeue.

When doing enqueue operator, grow the array by one and assign the new element to the last spot.

When do the dequeue operator, we need a counter variable C which initially is 0. The whole process is describe as follow.

(a) if dequeue array is empty, switch these two array and set C as 0.

(b) if C reach the end of the dequeue array, shrink dequeue array by one, decrease C by 1 and return.

(c) if C does not reach the end of the dequeue array, assign the last element to C^{th} position, shrink array by 1 and increase C by 1

Apparently, both dequeue enqueue operator cost constant number of array operation and the space usage is $O(n)$.

Problem 3 **Solution:**

$$S = 1 + c + c^2 + ... + n + cn$$
$$cS = c + c^2 + ... + cn + c^2n$$
$$(c-1)S = c^2n - 1$$
$$S = \frac{c^2n-1}{c-1}$$

Thus: the amortized time is

$$\frac{S}{n} = \frac{c^2n-1}{n(c-1)} \approx \frac{c^2}{c-1}$$

Problem 4 **Solution:**

Define potential function Φ = number of 9 in this number. When applied adding-one operation, if it does not carry this number, Φ at most increase 1, Otherwise if the number of bits flipped from 9 to 0 is k, then the actual time is k+1 and the Φ is reduced at least by k-1 Thus the amortized time is constant.