CSE373 Assignment 1

Aditya Balwani
SBUID: 109353920

April 4, 2016

1 Rotating Images

1.1 Part 1 : n is power of 2

```
# A: input matrix, n: width of matrix, power of 2
def rotateSquarePow2(A, n):
    if(n<=1):
        return A
    else:
        rotatedQuad1 = rotateSquarePow2(A[0:(n/2)-1][0:(n/2)-1], n/2)
        rotatedQuad2 = rotateSquarePow2(A[0:(n/2)-1][n/2:n-1], n/2)
        rotatedQuad3 = rotateSquarePow2(A[n/2:n-1][0:(n/2)-1], n/2)
        rotatedQuad4 = rotateSquarePow2(A[n/2:n-1][n/2:n-1], n/2)

        A[0:(n/2)-1][0:(n/2)-1] = rectangularCopy(rotatedQuad4)
        A[0:(n/2)-1][n/2:n-1] = rectangularCopy(rotatedQuad1)
        A[n/2:n-1][0:(n/2)-1] = rectangularCopy(rotatedQuad2)
        A[n/2:n-1][n/2:n-1] = rectangularCopy(rotatedQuad3)

        return A
```

1.2 Part 2 : n not is power of 2

```
# A: input matrix, m: height of A, n: width of matrix
def rotateArbitrary(A, m, n):
    if(n<=1):
        return A
    else:
        rotatedQuad1 = rotateArbitrary(A[0:(m/2)-1][0:(n/2)-1], m/2, n/2)
        rotatedQuad2 = rotateArbitrary(A[0:(m/2)-1][n/2:n-1], m/2, n/2)
        rotatedQuad3 = rotateArbitrary(A[n/2:m-1][0:(n/2)-1], m/2, n/2)
        rotatedQuad4 = rotateArbitrary(A[n/2:m-1][n/2:n-1], m/2, n/2)

        B[0:(m/2)-1][0:(n/2)-1] = rectangularCopy(rotatedQuad4)
        B[0:(m/2)-1][n/2:n-1] = rectangularCopy(rotatedQuad1)
        B[n/2:m-1][0:(n/2)-1] = rectangularCopy(rotatedQuad2)
        B[n/2:m-1][n/2:n-1] = rectangularCopy(rotatedQuad3)

        return A
```

1.3 Part 3: find T(n) if rectangular copy is $O(n^2)$

Assuming rc(a) is the running time of Rectangular Copy on a (a x a) matrix

$$
\begin{aligned}
T(n) &= 4T(n/2) + 4rc(n/2) + c' \\
&= 4^2T(n/2^2) + 4^2rc(n/2^2) + 4rc(n/2) + c' + c' \\
&= ... \\
&= 4^iT(n/2^i) + \sum_{j=1}^{i} 4^j rc(n/2^j) + ic'
\end{aligned}
$$

The base case is $T(1) \leq c$, in which $n/2^i = 1$. We have $2^i = n$ and $i = logn$. Since we know that $rc(a) = O(a^2)$, it means there exists $c_0 st rc(a) \leq c_a(a^2)$. Then we have

$$4^j rc(n/2^j) \leq 4^j(c_0(n/2^j)^2) = c_0(4^j(n/2^j)^2 = c_0 n^2)$$

Hence we have :

$$\begin{aligned}
T(n) &= 4^i T(n/2^i) + \sum_{j=1}^{i} 4^j rc(n/2^j) + ic' \\
&\leq cn^2 + \sum_{j=1}^{i} c_0 n^2 + ic' \\
&= cn^2 + logn(c_0 n^2) + ic' \\
&= O(n^2 logn)
\end{aligned}$$

1.4 Part 4: find T(n) if rectangular copy is $O(n)$

Assuming rc(a) is the running time of Rectangular Copy on a (a x a) matrix

$$\begin{aligned}
T(n) &= 4T(n/2) + 4rc(n/2) + c' \\
&= 4^2 T(n/2^2) + 4^2 rc(n/2^2) + 4rc(n/2) + c' + c' \\
&= ... \\
&= 4^i T(n/2^i) + \sum_{j=1}^{i} 4^j rc(n/2^j) + ic'
\end{aligned}$$

The base case is $T(1) \leq c$, in which $n/2^i = 1$. We have $2^i = n$ and $i = logn$. Since we know that $rc(a) = O(a^2)$, it means there exists $c_0 st rc(a) \leq c_a(a)$. Then we have

$$4^j rc(n/2^j) \leq 4^j(c_0(n/2^j)) = c_0(4^j(n/2^j) = c_0 n 2^j)$$

Hence we have :

$$\begin{aligned}
T(n) &= 4^i T(n/2^i) + \sum_{j=1}^{i} 4^j rc(n/2^j) + ic' \\
&= cn^2 c_0 \sum_{j=1}^{i} 2^j n + ic' \\
&= cn^2 + c_0(2^{i+1} - 2)n + ic' \\
&= cn^2 + c_0(2n - 2)n + ic' \\
&= (c + 2c_0)n^2 - 2c_0 n + c' logn \\
&= O(n^2)
\end{aligned}$$