**HW1**

Student ID: 40747044s

**1. Complexity**

For each function of the following C program, find out the tightest bound using notation. **Explanation or proof is needed.** Answers without any explanation or proof will **NOT** be given any credits.

(1).

void f(int n) {

int cnt = 0; 1

for ( int i=1; i<=n; i++ ) {  n+1

for ( int j=i+1; j<=n; j++ ) {  (n+2)(n-2)/2+1

cnt = cnt + 1; (n+2)(n-2)/2

}

}

}

Θ = n2

(2)

Void h(int n,int m)

{

int sum = 0; 1

for( inti=0;i<n;i++ ) {  n+1

sum = sum + i; n

}

for ( inti=0;i<m;i++ ) {  m+1

sum=sum+i; m

}

}

Θ = max(m,n)

(3). void g(int n)

{

if( n<=0 ){

return;

}

g(n-2);

g(n-1);

}

2T(g(n-2)) < T(g(n)) = T(g(n-2))+ T(g(n-1)) +1 < 2T(g(n-1))

2n/2 < T(g(n)) < 2n

Θ = 2n

(4)

Void j (int n)

{

if ( n == 1 ) {

return;

}

j(n/2);

j(n/2);

int total = 0;

for ( int i=0; i<n; i++ ) {

total = total + i;

}

}

2\*T(j(n/4))<T(j(n)) = T(j(n/2))+T(j(n/4)) < 2\*T(j(n/2))

n/2 < T(j(n)) < n

Θ = n

(5).

void f(int n) {

int cnt = 0;

for ( int i=1; i<=n; i++ ) {  n+1

for ( int j=i; j<=n; j+=i ) {  n(n+1)/2+1

cnt = cnt + 1; n(n+1)/2

}

}

}

Θ = n2

Hint: Squeeze theorem

**2. Fibonaci**

(1)

for( int i=0;i<n;i++ ) {  n+1

for( int j=0;j<n;j++ ) {  n(n+1)

c[i][j] = 0; n2

for( intk=0;k<n;k++){ n2(n+1)

c[i][j]=c[i][j]+a[i][k]\*b[k][j]; n3

}

}

}

Θ = n3

(2)

power(A):

ans = 1;

for( int i=0;i<N-1;i++ ) {  n+1

ans \*= A; n

}

return ans;

Θ = n

**3. Stack**

Grab(k):

while ( !stack.empty() && k > 0 ) {

Pop();

k = k-1;

}

now answer the following questions

(1). Let n be the number of elements in stack, what is the running time of operation: ? Using notation O with k or n (or both).

O = k

(2). Define a function: apply n random operations consecutively to a stack which is empty initially, what is the overall running time of this function? You should find out the bound using notation, and clearly describe your reason or proof.