21800612 임건호

**<Problem 1 – insertion sort>**

T(1) = 1  
T(N) = T(N-1) + N-1

T(N) = T(N-1) + N-1  
T(N-1) = T(n-2) + N-2

T(N-2) = T(n-4) + N-3

......  
T(2) = T(1) + 1

T(N) + T(N-1) + T(N-2) + T(N-3) + .... T(3) + T(2) =

T(N-1) + T(N-2) + T(N-3) + .... T(3) + T(2) + T(1) + N-1 + N-2 + N-3 + N-4 + … + 1

T(N) = T(1) + N-1 + N-2 + N-3 + … + 1 (open form)

T(N) = 1 + N(N+1)/2 (close form)

T(N) = O(N2) (big O)

**<Problem 2>**

T(1) = 1  
T(N)=T(N–1)+2 //2 is a constant like c

T(N) = T(N-1) + 2

T(N-1) = T(N-2) + 2

T(N-2) = T(N-3) + 2

......

T(2) = T(1) + 2

T(N) + T(N-1) + T(N-2) + T(N-3) + … + T(2) =

T(N-1) + T(N-2) + T(N-3) + T(N-4) + … + T(1) + 2 + 2 + … + 2

T(N) = T(1) + 2 + 2 + … + 2 = 2(N-1) (open form)

T(N) = 1 + 2(N-1) (closed form)

T(N) = O(N) (Big O)

**<Problem 3 – Power()>**

T(0) = 1  
T(n)=T(n–1)+1 // +1 is a constant

If we knew T(n−1), we could solve T(n).

T(n) = T(n–1) +1  
T(n – 1) = T(n – 2) + 1  
T(n – 2) = T(n – 3) + 1  
....  
T(2) = T(1) + 1  
T(1) = T(0) + 1

T(N) = 1 + 1 + … + 1

T(n) + T(n – 1) + T(n – 2) + … + T(2) + T(1) + T(0)

= T(n – 1) + T(n – 2) + T(n – 3) + … + 1 + 1 + 1

T(n) = 1 + 1 + … + 1 (Open form)

T(n) = 1 + N (Closed form)

T(n) = O(N) (Big O)

**<Problem4 – Power()>**

T(0) = 1  
T(1) = 1  
T(n) = T(n / 2) + 1 // Assume n is power of 2, +1 is a constant

T(0) = 1  
T(1) = 1  
T(n) = T(n/2) + 1

= T(n/4) + 1 + 1

= T(n/8) + 1 + 1 + 1

= T(n/16) + 1 + 1 + 1 + 1

....

= T(n/2k) + k

T(n) = T(n/2log2n) + log2n (Open form) in terms of n, 2k, k

= T(n/n) + log2n = T(1) + log2n (Open form)

= log2n (Closed form)

Therefore, T(n) = O(logn) (Big O)