資料結構作業

1.

a. 5n^2-6n = Θ(n^2)

(1) c1 = 4 , c2 = 5 , n0 = 6

* 0 <= 4 n^2 <= 5n^2-6n <= 5n^2 同除以n
* 4n <= 5n-6 <= 5n

(2) c1 = 3 , c2 = 6 , n0 = 3

* 0 <= 3 n^2 <= 5n^2-6n <= 6n^2 同除以n
* 3n <= 5n-6 <= 6n

b. n! = O(n^n)

(1) c = 1 , n0 = 0

* n! <= n^n 同除以n^n
* n!/n^n <= 1

(2) c = 2 , n0 = 0

* n! <= 2n^n 同除以n^n
* n!/n^n <= 2

c. 2n^2+nlogn = Θ(n^2)

(1) c1 = 2 , c2 = 3 , n0 = 1

* 2n^2 <= 2n^2+nlogn <= 3n^2
* 2n^2 <= 2n^2+nlogn
* 2n^2+nlogn <= 2n^2 + n^2
* nlogn <= n^2

(2) c1 = 1 , c2 = 4 , n0 = 1

* n^2 <= 2n^2+nlogn <= 4n^2
* n^2 <= 2n^2+nlogn
* 2n^2+nlogn <= 2n^2 + 2n^2
* nlogn <= 2n^2

2. Rewrite the matrix multiplication function of Program 1.20 using allocated arrays: A: m\*n; B: n\*p; C: m\*p. Then compute the performance by tabular method like Figure 1.4 shown in the textbook.

Void mult(int A[m][n], int B[n][p], int C[m][p]){ //Step per exec. Freq. Total

Int i, j, k; 0 0 0

for(i = 0;i < m;i++) 1 m+1 m+1

for(j = 0;j < p;j++){ 1 m\*(p+1) m\*p+m

C[i][j] = 0; 1 m\*p m\*p

for(k = 0;k <= n;k++) 1 m\*p(n+1) m\*p\*n+m\*p

C[i][j] += A[i][k] \* B[k][j]; 1 m\*p\*n m\*p\*n

}

}

Step count = m+1 + m\*p+m + m\*p + m\*p\*n+m\*p +m\*p\*n

= 2m\*p\*n + 3m\*p + 2m +1

* T(m,p,n) = O(m\*p\*n)