**PROJECT REPORT**

Team name: Yunfan Li, Haichao Zhang, Jingyuan Xu

Course: CS325-001

***Pseudocode***

**Algorithm 1**:

a) sum🡨0, result🡨 -INF  
 for i=0 to (size of array A-1)  
 for j=i+1 to (size of array A-1)  
 for k=i to j  
 sum🡨sum+a[k]  
 if sum>result  
 result🡨sum  
 sum🡨0

In this algorithm, we do “add two numbers together” for about times, “takes the max of two numbers” for about times.

**Algorithm 2**:

b) sum🡨0, result🡨 -INF  
 for i=0 to (size of array A-1)  
 b[i]=a[i]  
 for j=i+1 to (size of array A-1)   
 b[j]🡨b[j-1]+a[j]  
 for k=i to (size of array A-1)  
 if b[k]>result  
 result=b[k]

In this algorithm, we do “add two numbers together” for about times, “take the max of two numbers” for about times.

**Algorithm 3**:

c) All[110]={-INF}, END[110]={-INF}  
 All[0]=End[0]=a[0]  
 for i=0 to (size of array A)  
 End[i]=max of End[i-1]+a[i] and a[i]  
 All[i]=max of End[i] and All[i-1]

In this algorithm, we do “add two numbers together” for about times, “take the max of two numbers” for about times.

***Run-time Analysis***

a)

so algorithm a’s bounds is ()

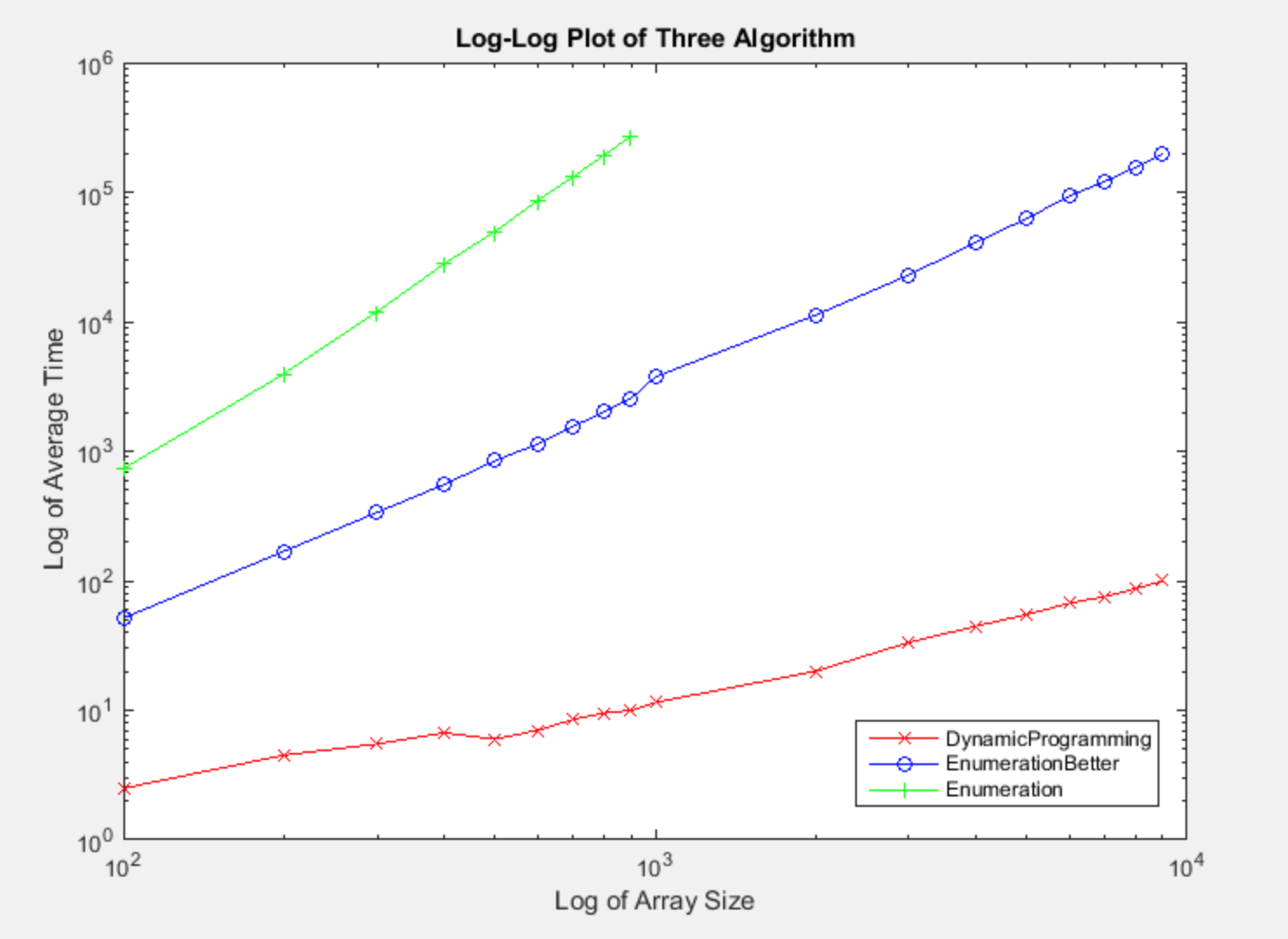
b)

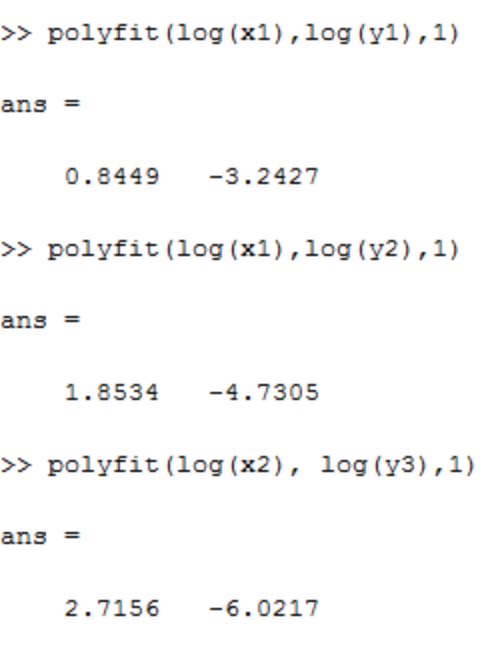
so algorithm b’s bounds is ()

c)

so algorithm c’s bounds is (n)

***Experimental run-time analysis***





All three algorithm’s experimental running time will be less than theoretical running time.

As for the first algorithm Enumeration plot, the theoretical running time is , the experimental running time: first loop is from i to n, second loop is from j=i+1 to n, third loop is from i to j, so the total time will be less than

As for the second algorithm Enumeration better plot, the theoretical running time is , the experimental running time: first loop is from i to n, second loop is from j=i+1 to n, so the total time will be less than

As for the third algorithm Dynamic Programming plot, the theoretical running time is n, the experimental running time: although the loop is from i to n, the computer’s calculating speed is faster than the expectation, so the time will be less than n.