Total: 2n+2

Ex-1.14: Matrix addition

Statement	s/e	Frequency	Total Steps
Void add(int a[][MAX_SIZE]...)	0	0	0
{	0	0	0
int i,j;	0	0	0
for(i=0;i<rows;i++)	1	rows+1	rows+1
for(j=0;j<cols;j++)	1	rows.(cols+1)	rows.cols+rows
c[i][j]=a[i][j]+b[i][j];	1	rows.cols	rows.cols
}	0	0	0
Total			2rows.cols+2.rows+1

Summary

The time complexity of a program is given by the number of steps taken by the program to compute the function it was written for. The no. of steps is itself a function of the instance characteristics. While any specific instance may have several characteristics (eg: the number of inputs, the no. of outputs, the magnitudes of the inputs and outputs, etc.), the no. of steps is computed as a function of some subset of these.

For example, we might wish to know how the computing (or run) time (ie. time complexity) increases as the number of inputs increase. In this case the number of steps will be computed as a function of the number of inputs alone. For a different program, we might be interested in determining how the computing time increases as the magnitude of one of the inputs increases. In this case the number of steps will be computed as a function of the magnitude of this input alone. Thus, before the step count of a program can be determined, we need to know exactly which characteristics of the problem instance are to be used.