Ex 1.18 (Complexity of matrix addition)

Statement	Asymptotic complexity
`void add(int a[][MAX_SIZE].....)`	0
`{`	0
`int i,j;`	0
`for(i=0; i<rows; i++)`	$\Theta(rows)$
`for(j=0; j<cols; j++)`	$\Theta(rows \cdot cols)$
`c[i][j]=a[i][j]+b[i][j];`	$\Theta(rows \cdot cols)$
`}`	0
Total	$\Theta(rows \cdot cols)$

Ex.1.19: (Time Complexity of Binary Search)

- The instance characteristic we shall use is the number n of elements in the list.
- Each iteration of the while loop takes $\Theta(1)$ time. We can show that the while loop is iterated atmost $\lceil \log_2(n+1) \rceil$ times.
- Since an asymptotic analysis is being performed, we don't need such an accurate count of the worst-case number of iterations. Each iteration except for the last results in a decrease in the size of the segment of list that has to be searched by a factor of about 2. That is, the value of right-left+1 reduces by a factor of about 2 on each iteration. So, this loop is iterated $\Theta(\log n)$ times in the worst case. As each iteration takes $\Theta(1)$ time, the overall worst case complexity of binsearch is $\Theta(\log n)$. Notice that the best case complexity is $\Theta(1)$ as in the best case searchnum is found in the 1st iteration of the while loop.