

**Better\_Enumeration:****Theoretical Run-time Analysis:****pseudo-code**

```
better_enumeration(a[1...n])
  For i=1...n
    Sum = 0
    For j =1..n
      sum=sum+a[j]
      If sum>maxsum
        Maxsum=sum
  return maxsum
```

The i outer loop runs from 1 to n, the inner loop j runs from 1 to n. Thus, the theoretical run-time is  $O(n^2)$ .

**Linear\_time:****Theoretical Run-time Analysis:****pseudo-code**

```
linear_time(a[1...n])
int b[] = a[]; //b is a clone of a
int max=a[0];
for i from 0 to n
  if(b[i-1]>0)
    b[i]=b[i-1]+a[i];
maxsum=max value in array b
return maxsum
```

The i loop track the input array elements from 1 to n.  $\sum_{i=1}^n O(1) = n * O(1) = O(N)$ , the theoretical run-time is  $O(N)$ .