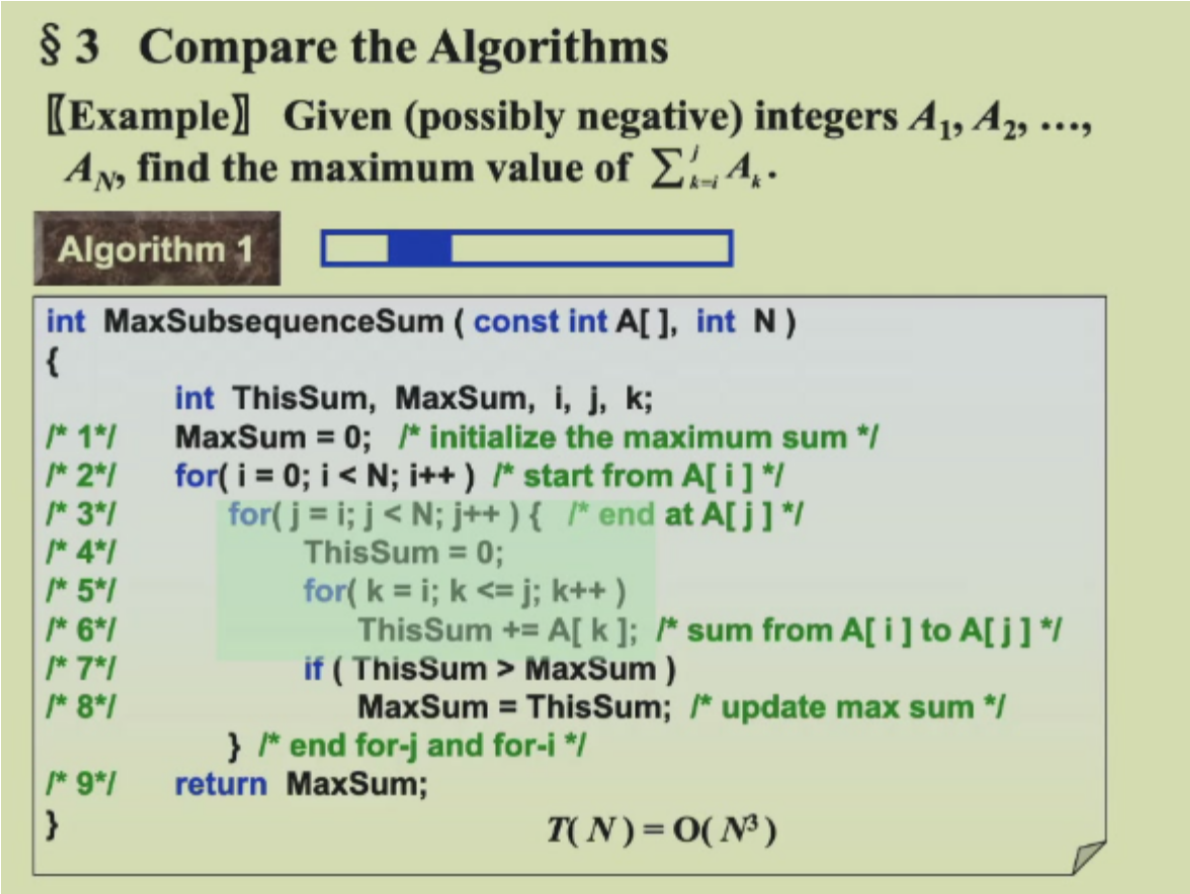
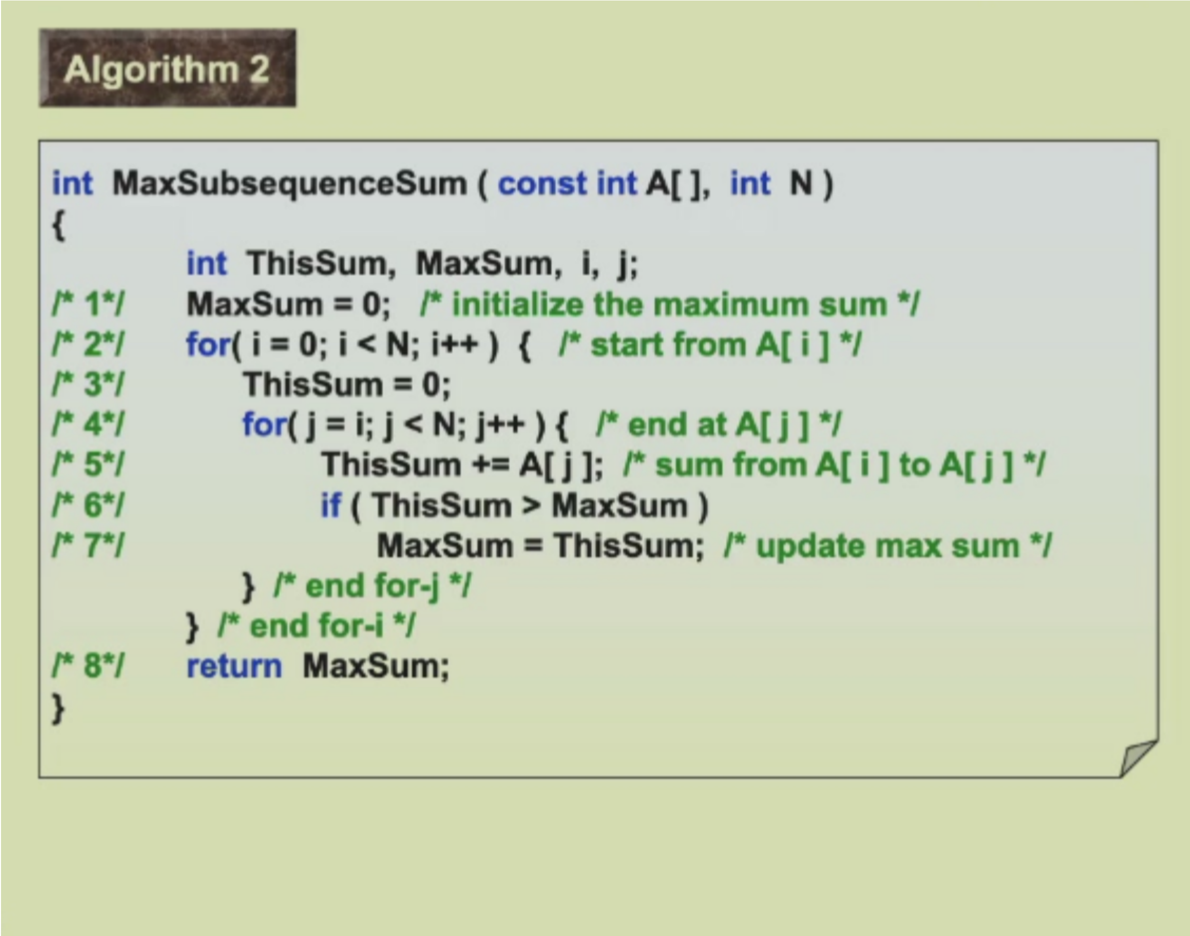
Analysis Algorithm

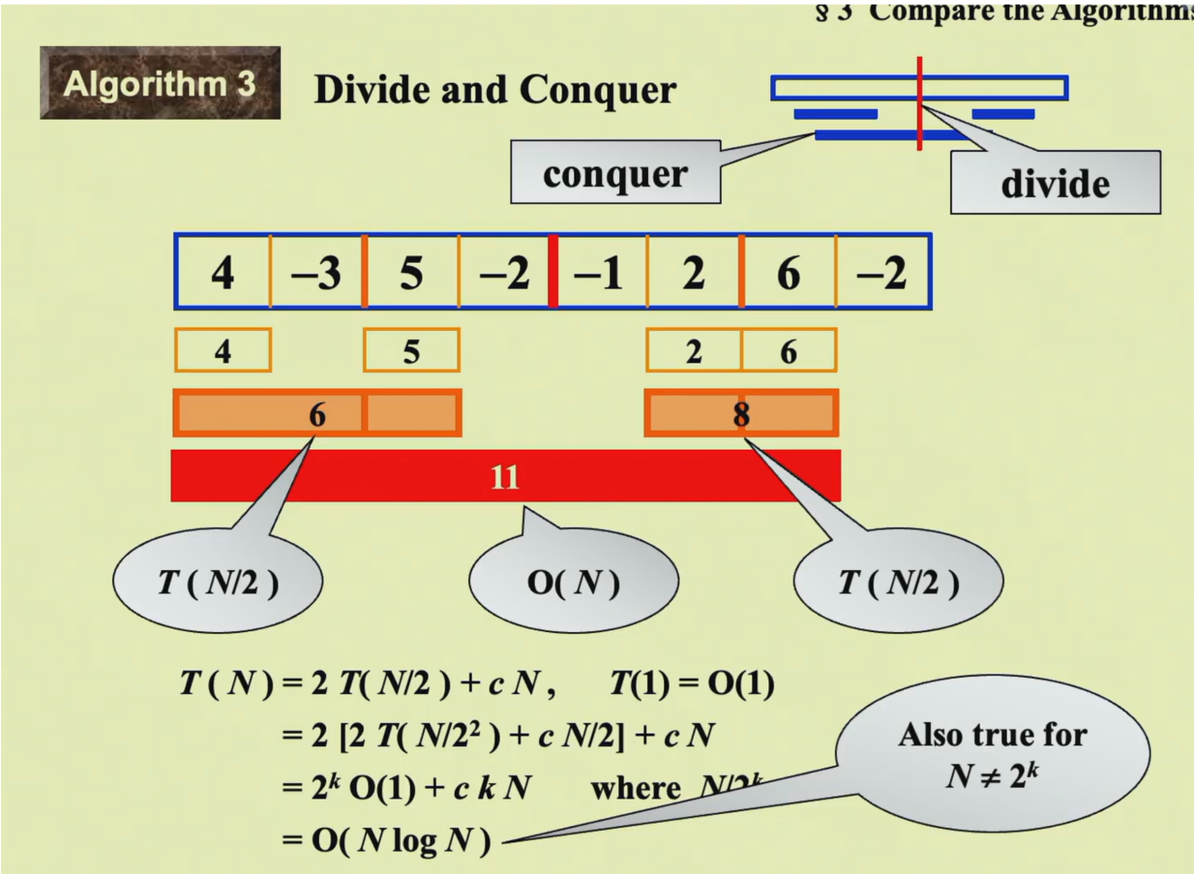
[Example] Sum from i to j



- 记忆前项和



- Divide and Conquer（分治法）

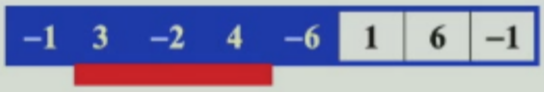


如何解T(N)：直接带入

- Online Algorithm

Algorithm 4 **On-line Algorithm**

```c
int MaxSubsequenceSum( const int  A[ ],  int  N )
{
            int  ThisSum, MaxSum, j;
/* 1*/      ThisSum = MaxSum = 0;
/* 2*/      for ( j = 0; j < N; j++ ) {
/* 3*/          ThisSum += A[ j ];
/* 4*/          if  ( ThisSum > MaxSum )
/* 5*/              MaxSum = ThisSum;
/* 6*/          else if ( ThisSum < 0 )
/* 7*/              ThisSum = 0;
            } /* end for-j */
/* 8*/      return MaxSum;
}
```

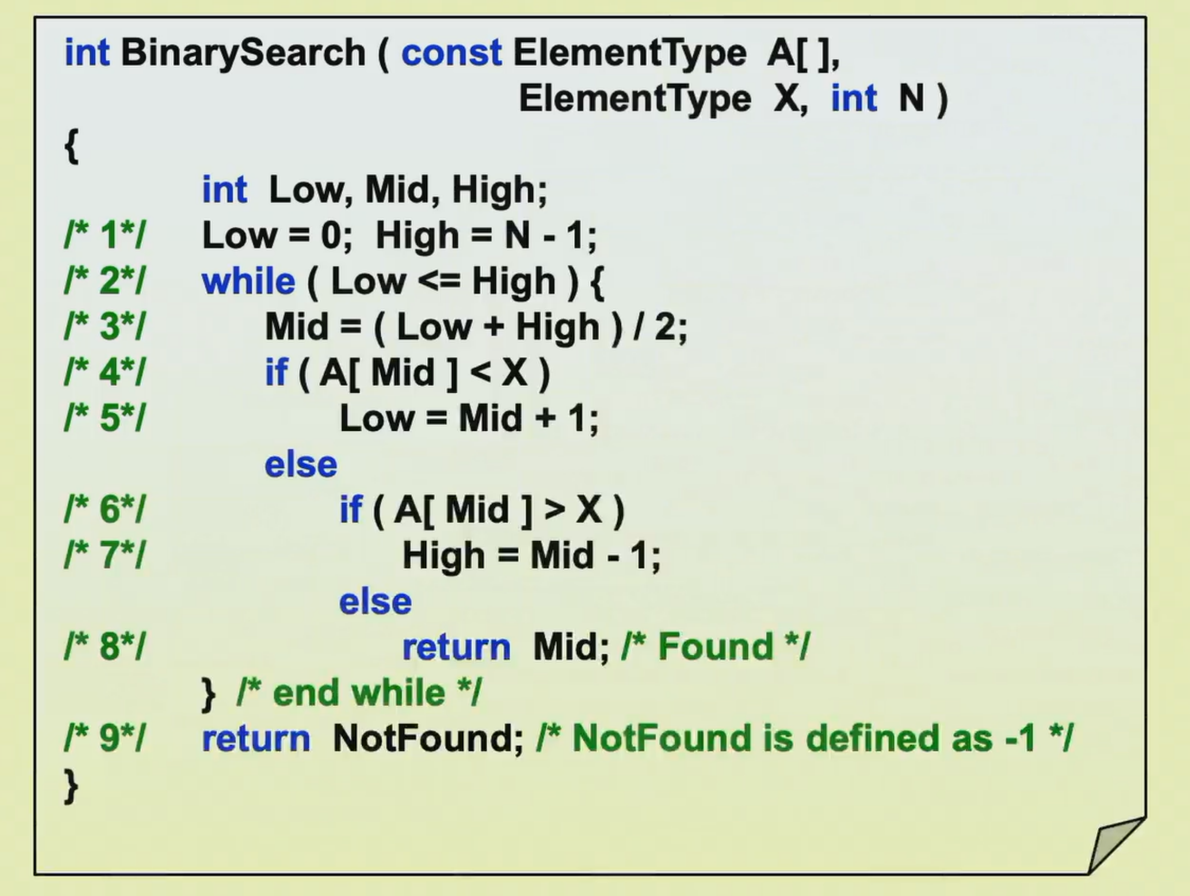


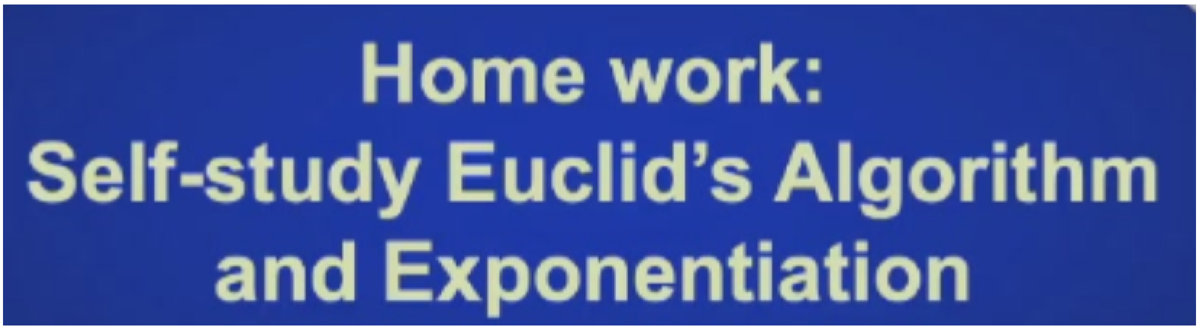
$T(N) = O(N)$

A[ ] is scanned once only.

> 负值不可能产生更大的和，因此置零重新计算
>
> 意义：
>
> 1. 快
> 2. 空间复杂度低
> 3. 随时停止即为当前最佳结果

[Example] Binary search





> 辗转相除法
>
> 指数算法（分治+递归）

Check your Analysis

§ 5 Checking Your Analysis

Method 1

When $T(N) = O(N)$, check if $T(2N)/T(N) \approx 2$

When $T(N) = O(N^2)$, check if $T(2N)/T(N) \approx 4$

When $T(N) = O(N^3)$, check if $T(2N)/T(N) \approx 8$

… …

Method 2

When $T(N) = O(f(N))$, check if

$$\lim_{N\to\infty} \frac{T(N)}{f(N)} \approx \text{Constant}$$

Read the example given on p.28 (Figures 2.12 & 2.13).