分治法:

概觀：

分治法(Divide and Conquer)是一種常用的演算手法，它的基本精神在於把問題拆分成較小規模的子問題，逐一解決後再合併結果來解決規模較大的問題。

複雜度分析：

運用到分治法的演算法恆河沙數，某些情況下，它的時間複雜度 T(n)能夠滿足如下遞迴關係：

$$T(n) = a\,T(n/b) + f(n), \text{ where } a \geqq 1, b > 1$$

我們可以利用主定理(Master Theorem)嘗試對 T(n)進行分析

考慮情況一：

$$f(n) \in O(n^c), \text{for } c < log_b^a$$

那麼：

$$T(n) \in \theta(n^{log_b^a})$$

考慮情況二：

$$f(n) \in \theta\left(n^{log_b^a} log^k n\right), \text{for constant } k \geq 0$$

那麼：

$$T(n) \in \theta(n^{log_b^a} log^{k+1} n)$$

考慮情況三：

$$f(n) \in \Omega(n^c), \text{for } c > log_b^a$$

那麼：

$$T(n) \in \theta(f(n))$$

實例一： 排序

快速排序法：(Quick Sort)

```
template<typename iter>
void quickSort(iter begin, iter end){
	static int randGenerated = 0;
	int size = end-begin;
	if(size<=1) return;
	if(!randGenerated) srand(time(NULL));
	randGenerated = (randGenerated+1)%50000;
	swap(begin[0],begin[rand()%size]);
	iter form = begin+1;
	iter latt = begin+size-1;
	while(form<=latt)
		if(*form<*begin)
			++form;
		else if(*begin<*latt)
			--latt;
		else
			swap(*form,*(latt--));
	swap(*begin,*latt);
	quickSort(begin,latt);
	quickSort(form,end);
}
```

合併排序法：(Merge Sort)

```
template<typename iter>
void mergeSort(iter begin, iter end){
	if(begin+1>=end) return;
	iter mid = begin+(end-begin)/2;

	/**Divide**/
	mergeSort(begin,mid);
	mergeSort(mid,end);

	/**Merge**/
	iter form = begin;
	iter latt = mid;
	typename iterator_traits<iter>::value_type tempArr[end-begin];
	for(int cnt=0;cnt<sizeof(tempArr);cnt++)
		if(form!=mid && latt!=end)
			if(*form<*latt)
				tempArr[cnt] = *(form++);
			else
				tempArr[cnt] = *(latt++);
		else if(form != mid)
			tempArr[cnt] = *(form++);
		else if(latt != end)
			tempArr[cnt] = *(latt++);
	for(iter cnt=begin; cnt!=end ; ++cnt)
		*cnt = tempArr[cnt-begin];
}
```

實例二：大數字乘法(Big Integer Multiplication)

實例三：凸包探詢(Convex Hull)