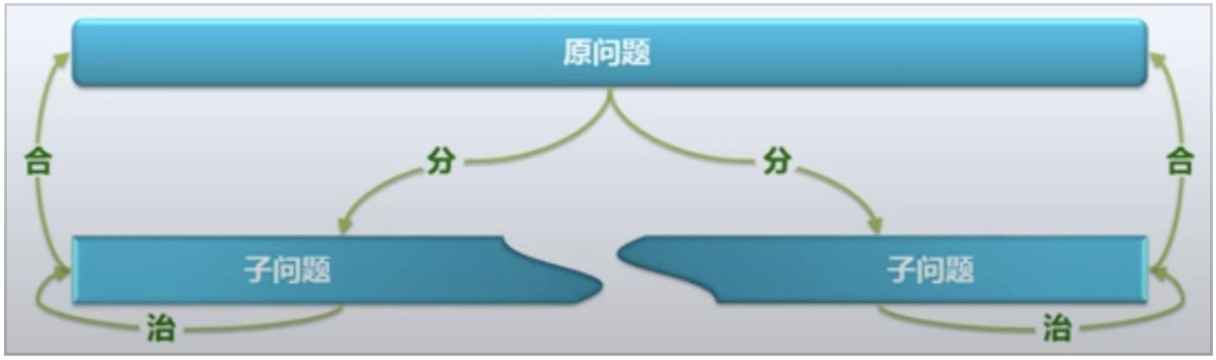
01-E-5 分而治之 & 01-E-6 数组求和-二分递归

#数据结构邓神

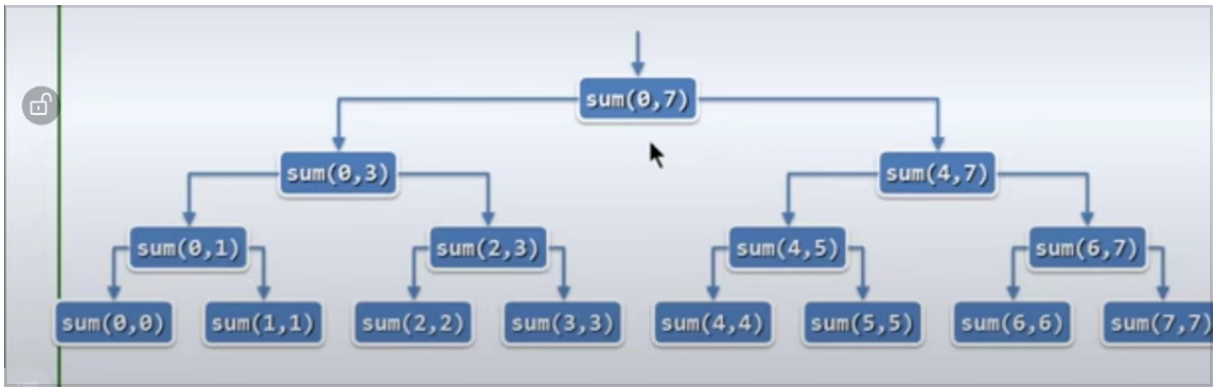
divide-and -conquer

为了求解一个大规模的问题，可以将其划分为若干（通常两个）字问题，规模大体相当

分别求解字问题，从而从子问题的解得出原来问题的解




```
int sum(int A[],int lo,int hi){ // 区间范围 [lo,hi]
    if (lo == hi) return A[lo];
    int mi = (lo + hi) >> 1; // >> 1 等价于 /2
    return sum(A,lo,mi) + sum(A,mi+1,hi);
} // 入口形式为 sum(A,0,n-1)
```



T(n) = 各层递归实例所需时间之和 // 递归跟踪

= O(1) * (2^0 + ... + 2^logn)

= O(1) * (2^log(n+1) - 1) = O(n)

从递归的角度来看：为求解sum(A,lo,hi),需递归求解sum(A,lo,hi) // 2 * T(n/2)

进而将字问题的解累加 // O(1)

递归基 : sum(A,lo,lo) // O(1)

递归关系: T(n) = 2*T(n/2) + O(1)

T(1) = O(1)

最后 T(n) = O(n)