凡治众如治寡，分数是也

The control of a large force is

the same principle as

the control of a few men:

it is merely a question of

dividing up their numbers.

1.绪论

迭代与递归

分而治之

邓俊辉

deng@tsinghua.edu.cn

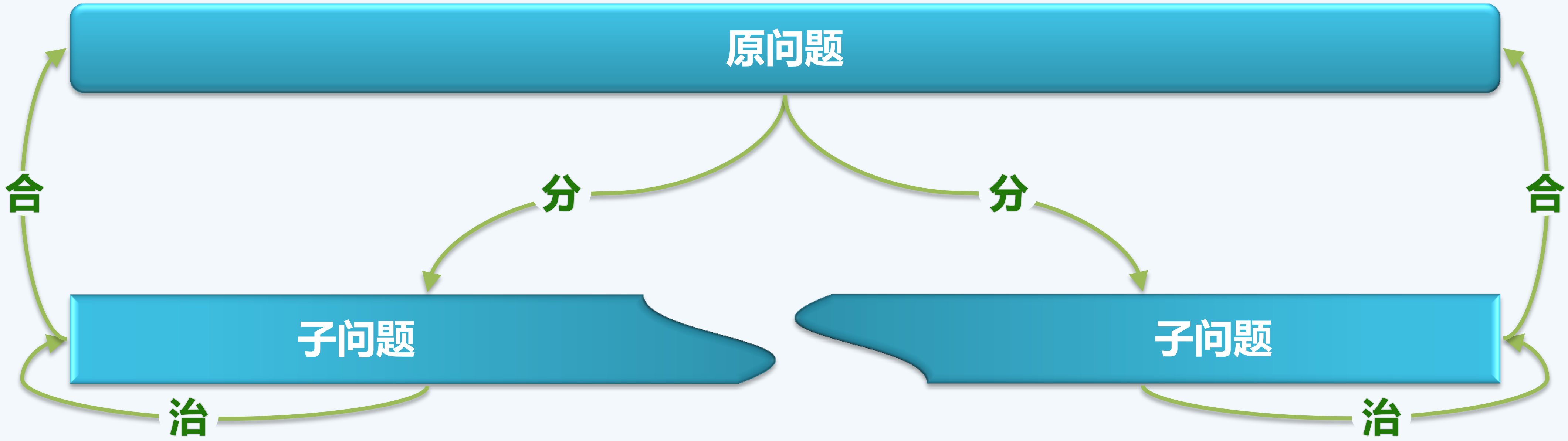
Divide-and-conquer

❖ 为求解一个大规模的问题，可以

将其划分为若干（通常两个）子问题，规模大体$\boxed{\text{相当}}$

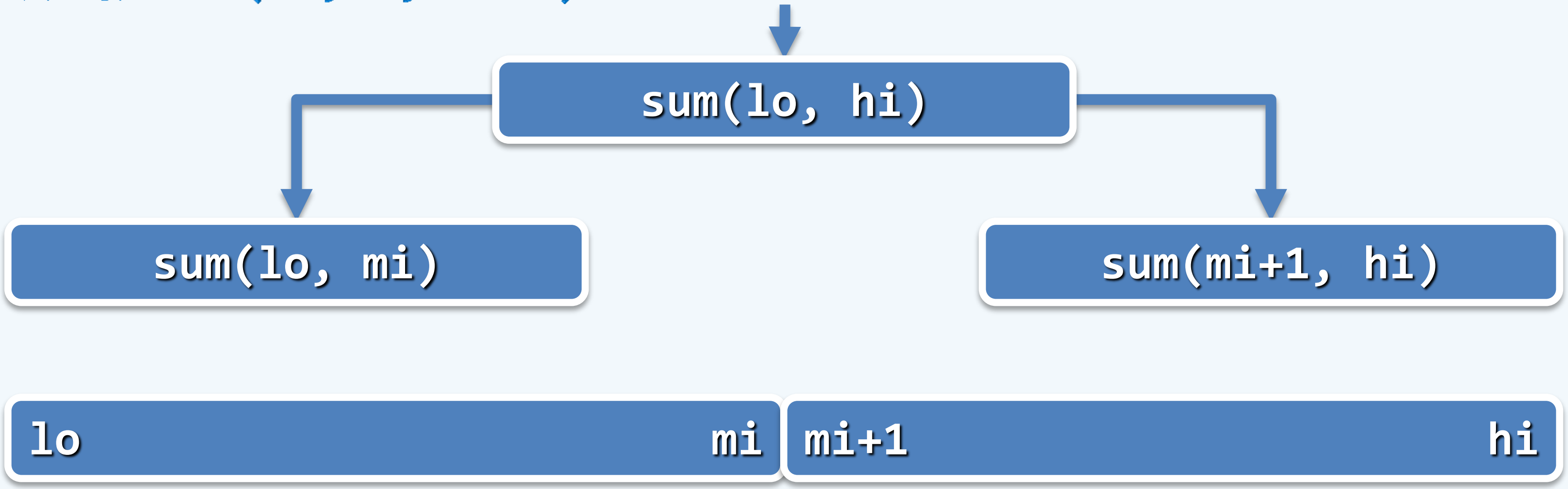
分别求解子问题

由子问题的解，得到原问题的解



Binary Recursion

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

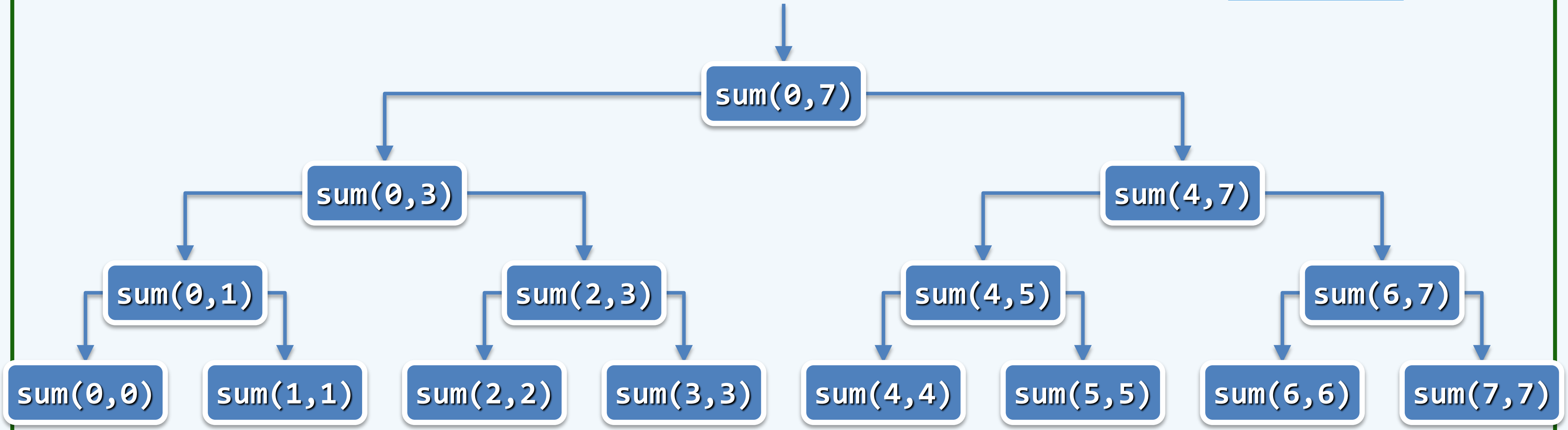


Binary Recursion: Trace

❖ $T(n)$ = 各$\boxed{\text{层}}$递归实例所需时间之$\boxed{\text{和}}$ //递归跟踪

$= \mathcal{O}(1) \times (2^0 + 2^1 + 2^2 + \dots + 2^{\log n})$

$= \mathcal{O}(1) \times (2^{1 + \log n} - 1) = \mathcal{O}(n)$ //更快捷地，几何级数与$\boxed{\text{最大/末项}}$等阶



Binary Recursion: Recurrence

❖ 从递推的角度看，为求解sum(A, lo, hi)，需

- 递归求解sum(A, lo, mi)和sum(A, mi+1, hi)，进而 //2*T(n/2)
- 将子问题的解累加 //$\mathcal{O}(1)$

❖ 递推关系 $\boxed{T(n) = 2*T(n/2) + \mathcal{O}(1)}$

$\boxed{T(1) = \mathcal{O}(1)}$ //base: sum(A, k, k)

❖ 求解

$$T(n) = 2*T(n/2) + c_1$$

$$T(n) + c_1 = 2*(T(n/2) + c_1) = 2^2 * (T(n/4) + c_1) = \ldots$$

$$= 2^{\log n}(T(1) + c_1) = n*(c_2 + c_1)$$

$$T(n) = (c_1+c_2)n - c_1 = \boxed{\mathcal{O}(n)}$$