Parallel Computing

Chapter 7 Supplement

Fundamental Techniques of Designing Parallel Algorithms

Spring, 2016

基本设计技术

- **7.1 Partitioning Principle**
- **7.2 Divide-and-Conquer Strategy**
- **7.3 Balanced Trees Method**
- **7.4 Doubling Techniques**
- **7.5 Pipelining Techniques**

7.1 划分设计技术

- **设计思想**
  - 将原问题划分成p个独立的几乎相等的子问题;
  - p台处理器并行地求解各子问题。

Remark:

  - 划分重点在于:子问题易解,组合成原问题的解方便;
  - 有别于分治法

- **常见划分方法**
  - 均匀划分
  - 方根划分
  - 对数划分
  - 功能划分(补)

7.1.4 功能划分

- 方法: n个元素A[1..n]分成等长的p组，每组满足

  某种特性。

- 示例: (m, n)选择问题(求出n个元素中前m个最小者)

  - 功能划分：要求每组元素个数必须大于m；

  - 算法是基于Batcher排序网络，下面先介绍一些预备知识
:

  1.Batcher比较器

  2.奇偶归并及排序网络:

  网络构造、奇偶归并网络、奇偶排序网络

  3.双调归并及排序网络:

  定义与定理、网络构造、双调归并网络、双调排序网络

7.1.4 功能划分

1. Batcher比较器

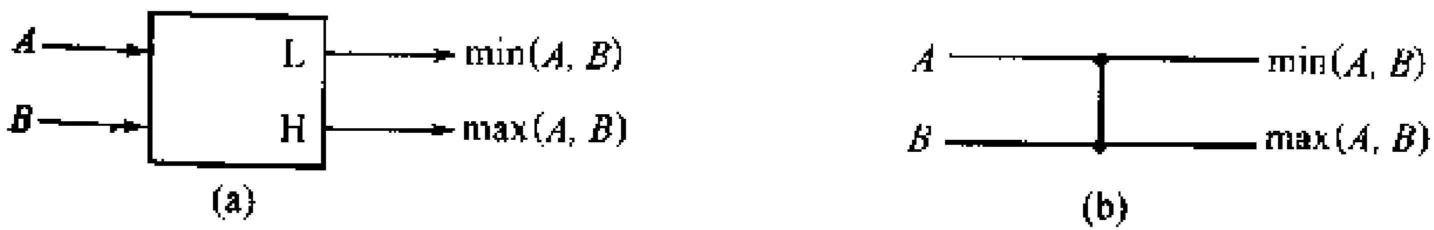


图 3.1 Batcher 比较器

- 比较和条件交换操作: CCI
- 比较器网络：用Batcher比较器连成完成某一功能的网络
- 假定：每次每个元素只能与另一个元素比较
- 比较器网络的参数：比较器数目、延迟级数

7.1.4 功能划分

2.1奇偶归并网络构造

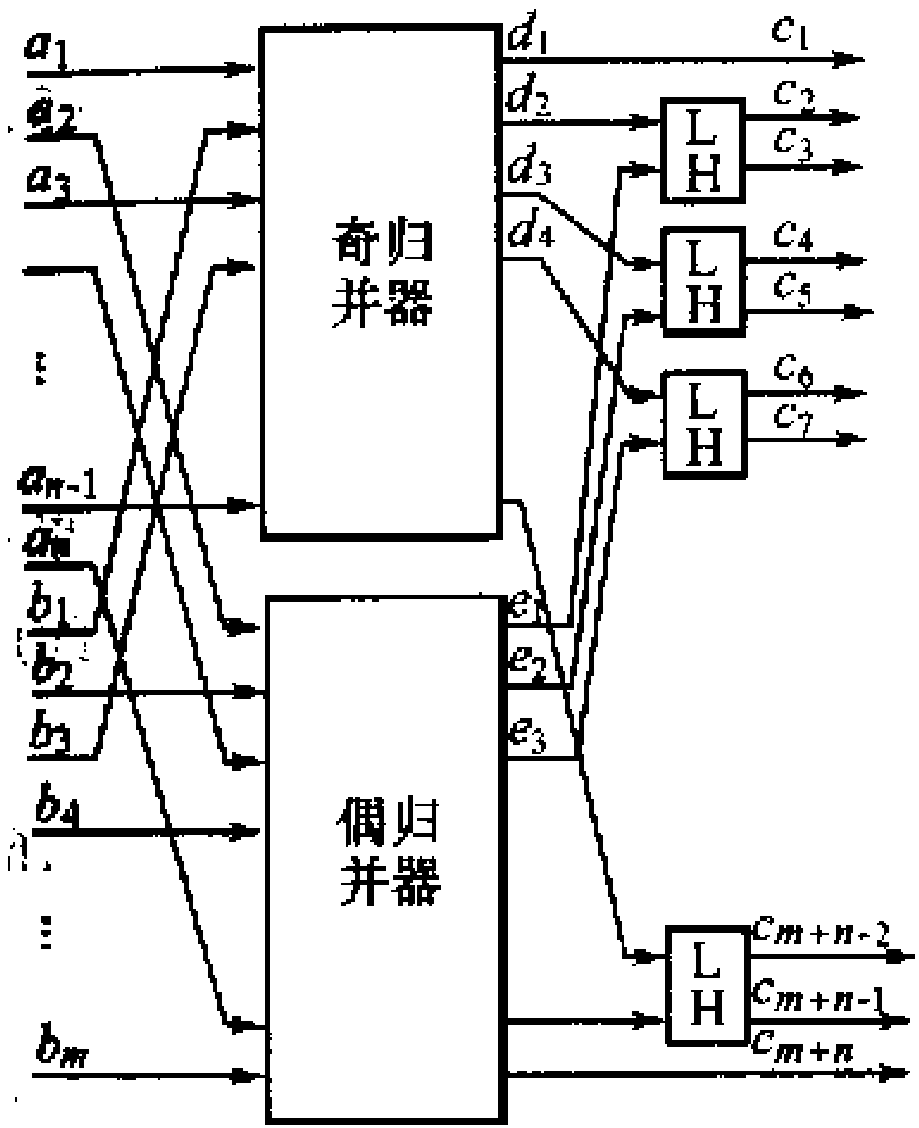
- 有序序列A:$a_1,a_2,\cdots,a_n$

  B: $b_1,b_2,\cdots,b_m$

- 归并思想：
  - A, B中奇数号元素进入奇归并器；
  - A, B中偶数号元素进入偶归并器；
  - 再将奇归并器与偶归并器的输出交叉比较

注: (m,n)规模划分为：

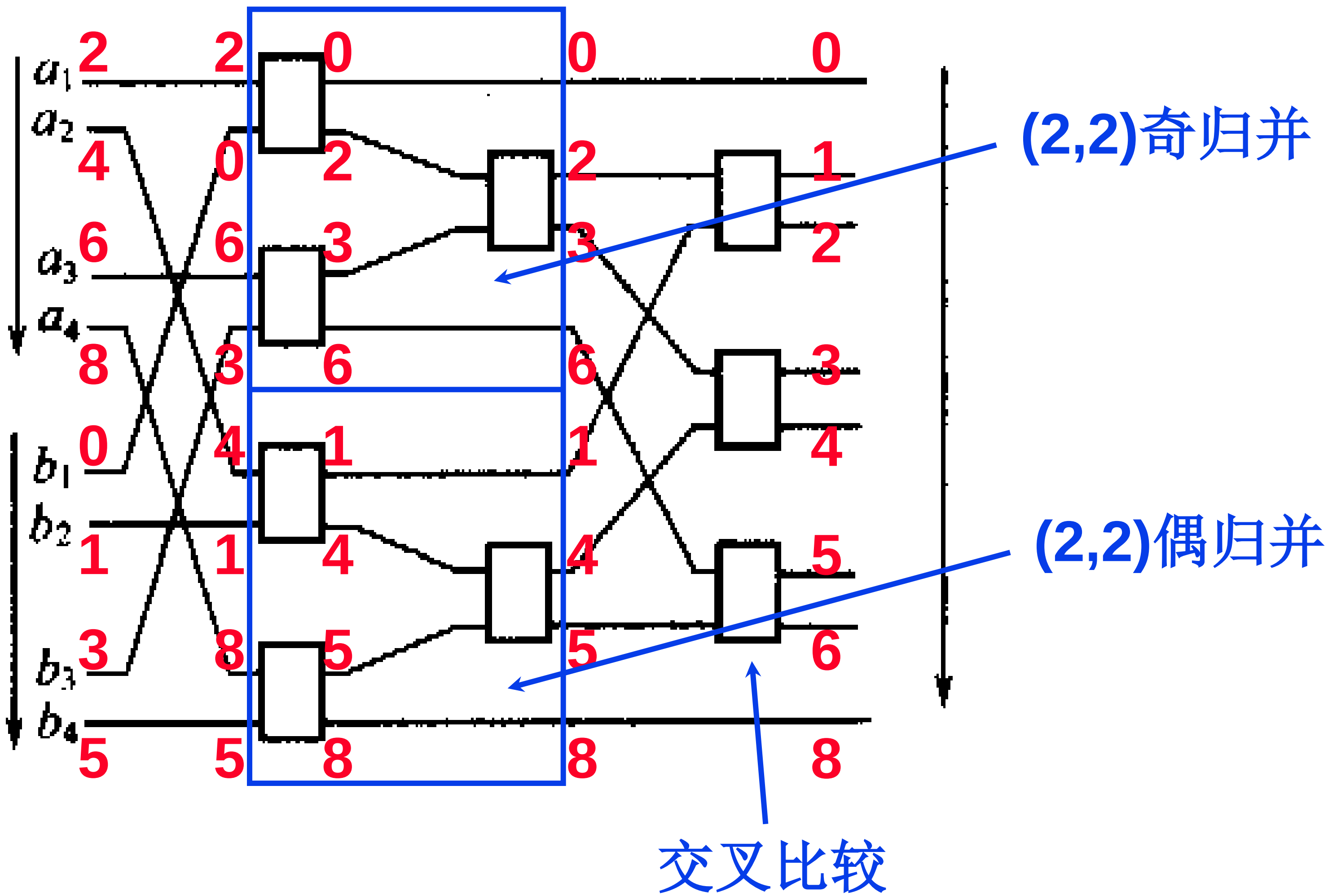
$$\begin{cases}(\lceil m/2\rceil,\lceil n/2\rceil)\text{奇}\\(\lfloor m/2\rfloor,\lfloor n/2\rfloor)\text{偶}\end{cases}$$



7.1.4 功能划分

2.2 奇偶归并示例：m=n=4 A=(2,4,6,8) B=(0,1,3,5)

(4, 4)→2×(2, 2)→4×(1, 1)



.1.4 功能划分

2.3 奇偶排序网络

- 基于奇偶归并网络
- 示例： **B(8)**

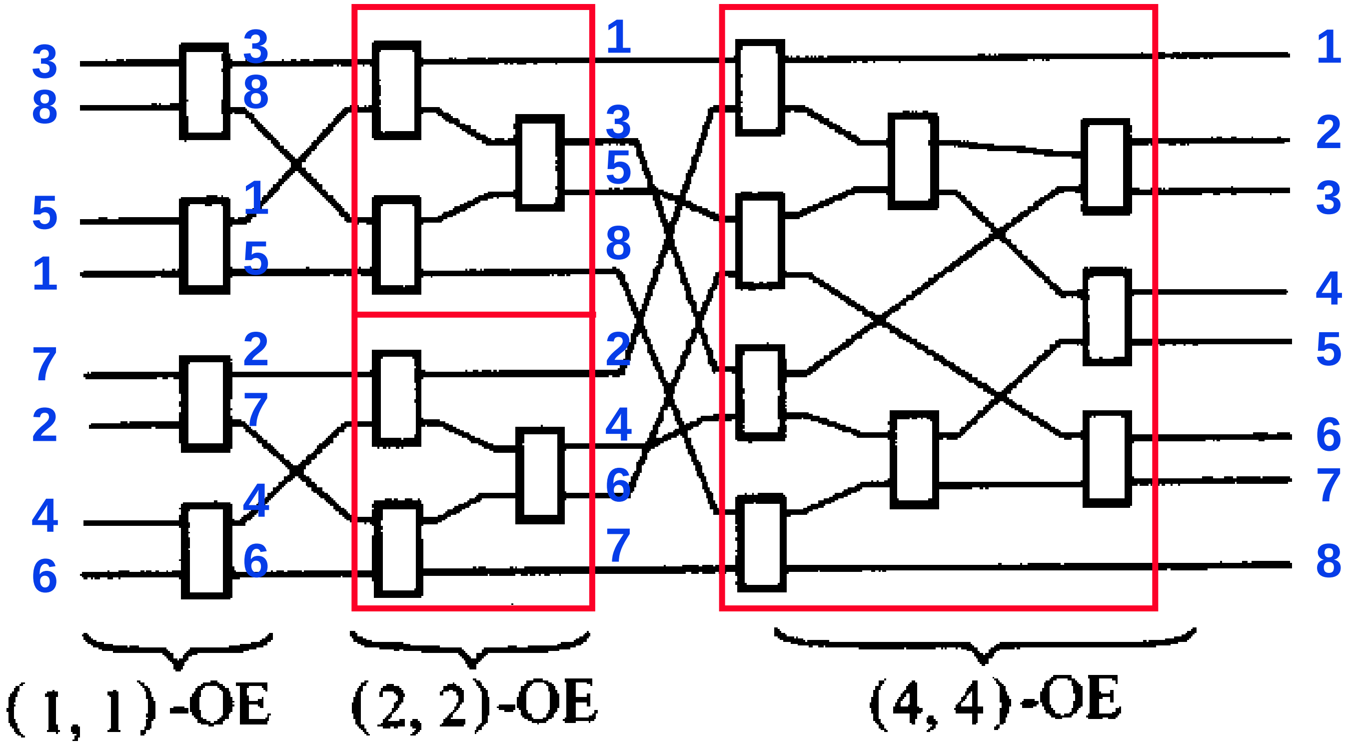


图 3.6 8 输入的奇偶排序网络

7.1.4 功能划分

3.1 定义及定理

- 定义: 一个序列$a_1,a_2,\cdots,a_n$是双调序列(Bitonic Sequence)，如果：

  (1)存在一个$a_k(1\leqslant k\leqslant n)$, 使得$a_1\geq\cdots\geq a_k\leq\cdots\leq a_n$成立；或者

  (2)序列能够循环移位满足条件(1)

- 示例：

  序列(1,3,5,7,8,6,4,2,0), (7,8,6,4,2,0,1,3,5)和(1,2,3,4,5,6,7,8)

  都是双调序列。 a_k

- 定理(Batcher定理)：

  设序列$a_1,\cdots,a_n,a_{n+1},\cdots, a_{2n}$是一个双调序列, 记

  $b_i=\min\{a_i, a_{i+n}\}$ ==> $MIN=\{b_1,\cdots,b_n\}$,

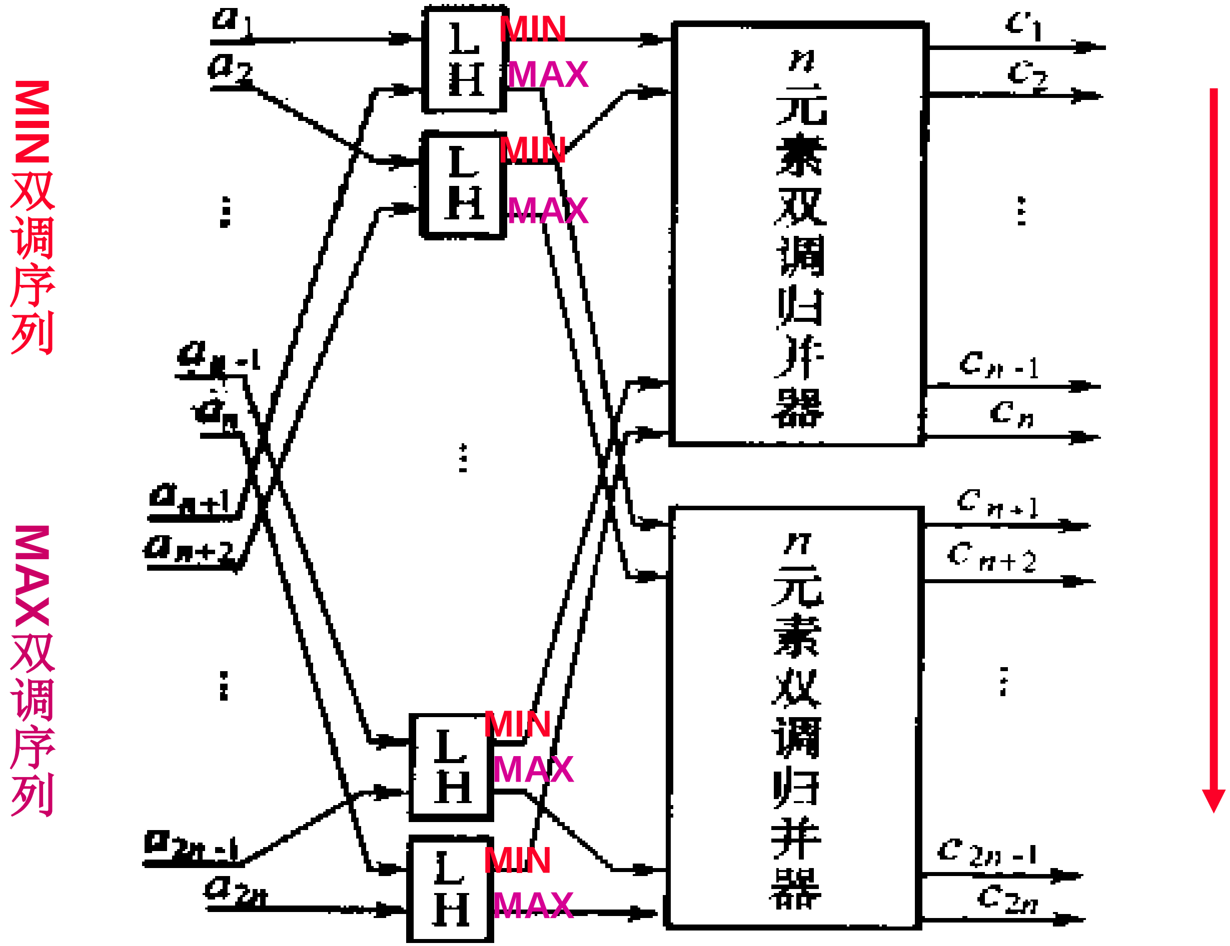
  $c_i=\max\{a_i, a_{i+n}\}$ ==> $MAX=\{c_1,\cdots,c_n\}$, 则

  (1) $b_i\leqslant c_j$ $(1\leq i, j\leq n)$ (2) MIN和MAX序列仍是双调的

7.1.4 功能划分

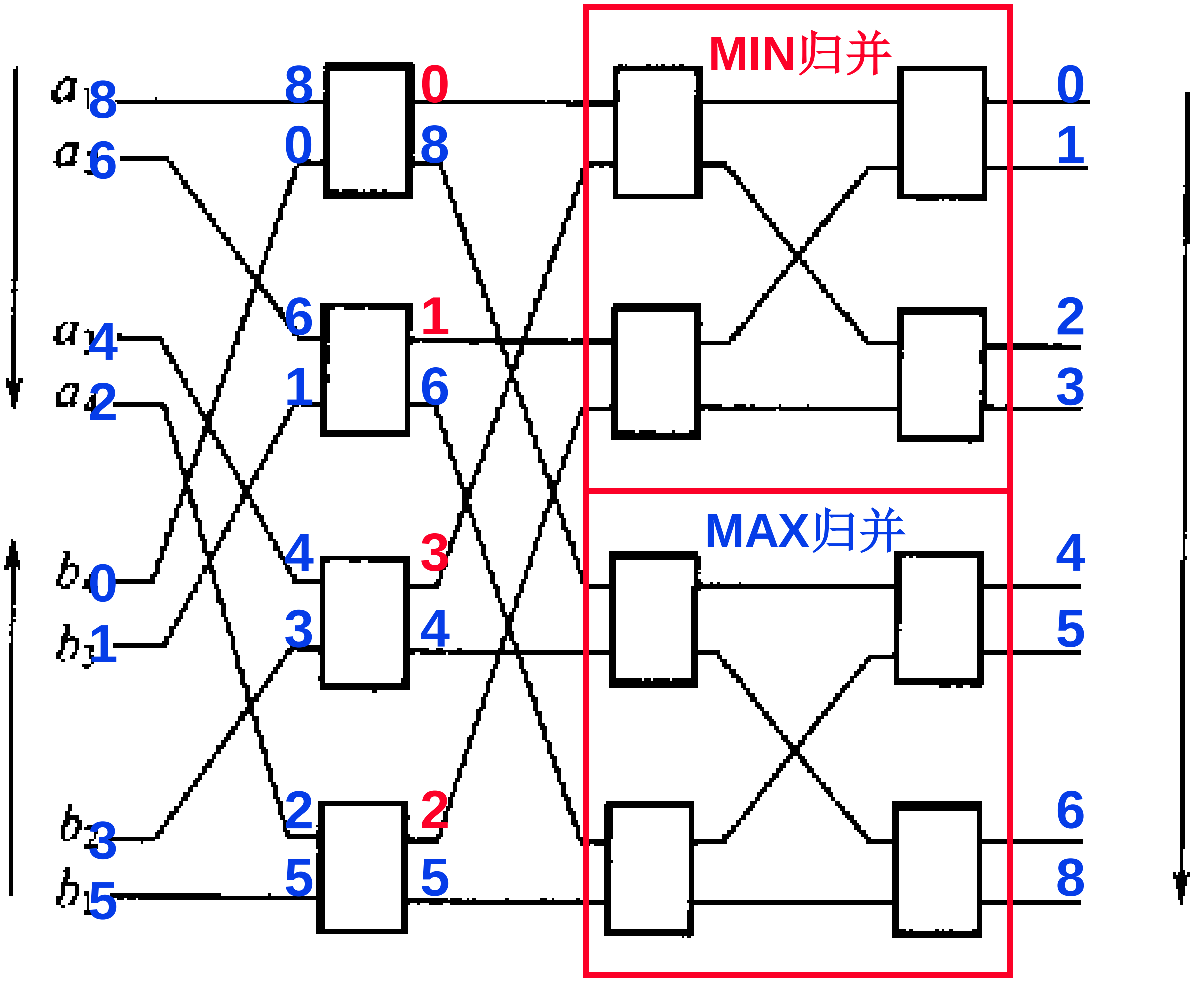
3.2 双调归并网络构造(依据Batcher定理)

- 2n个输入的双调序列两两比较形成2个大小为n的MIN和MAX序列
- MIN和MAX序列是双调的，可以递归重复进行下去



7.1.4 功能划分

3.3 双调归并示例:双调序列(8,6,4,2,0,1,3,5)的(4,4)双调归并网络



两两比较　　2个(2,2)双调归并网络

7.1.4 功能划分

3.4 双调排序网络

- 基于双调归并网络
- 示例：**B(8)**

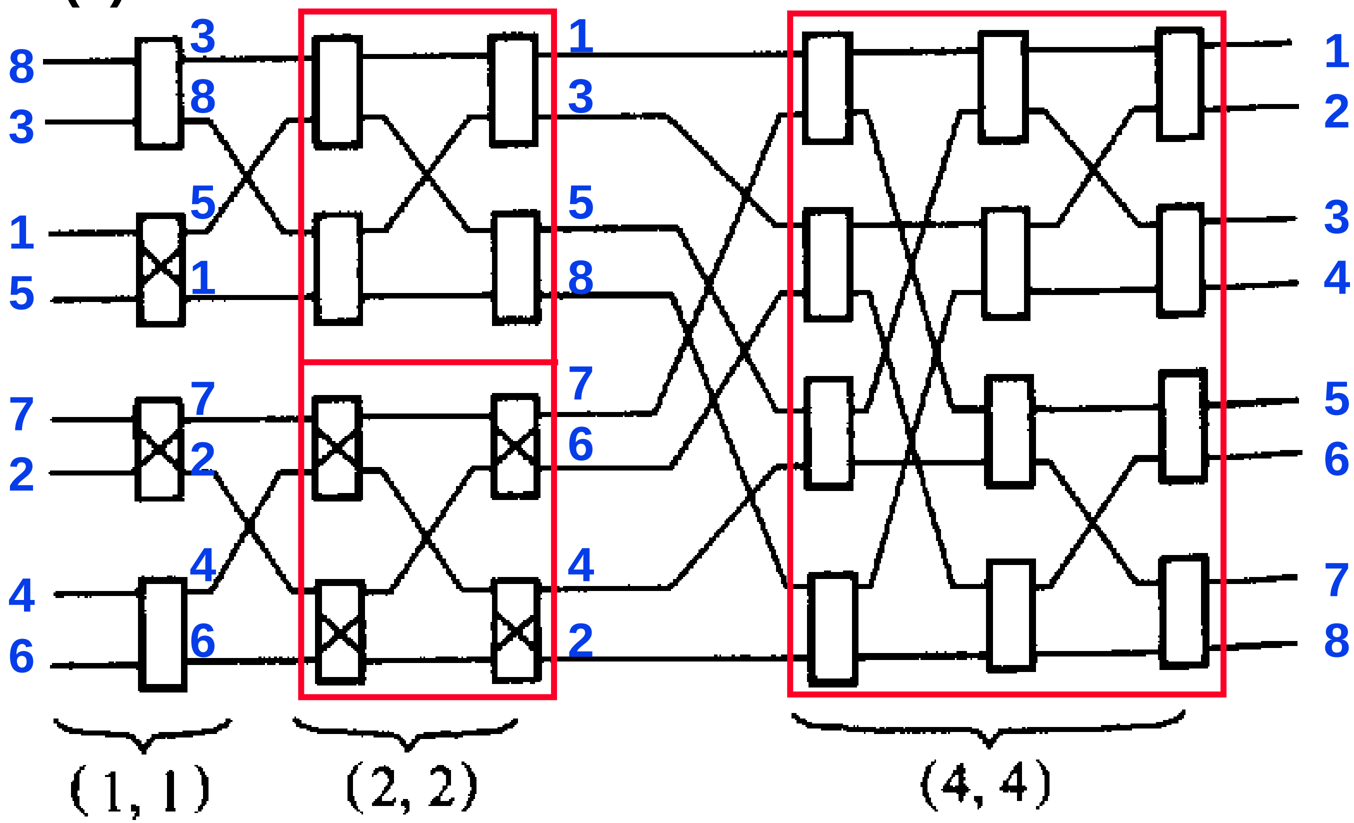


图 3.7　8 输入的双调排序网络

7.1.4 功能划分

- 基于划分原理的(m,n)-选择过程

①将n个输入数据划分成若干个大小相等的子序列(≥m)；

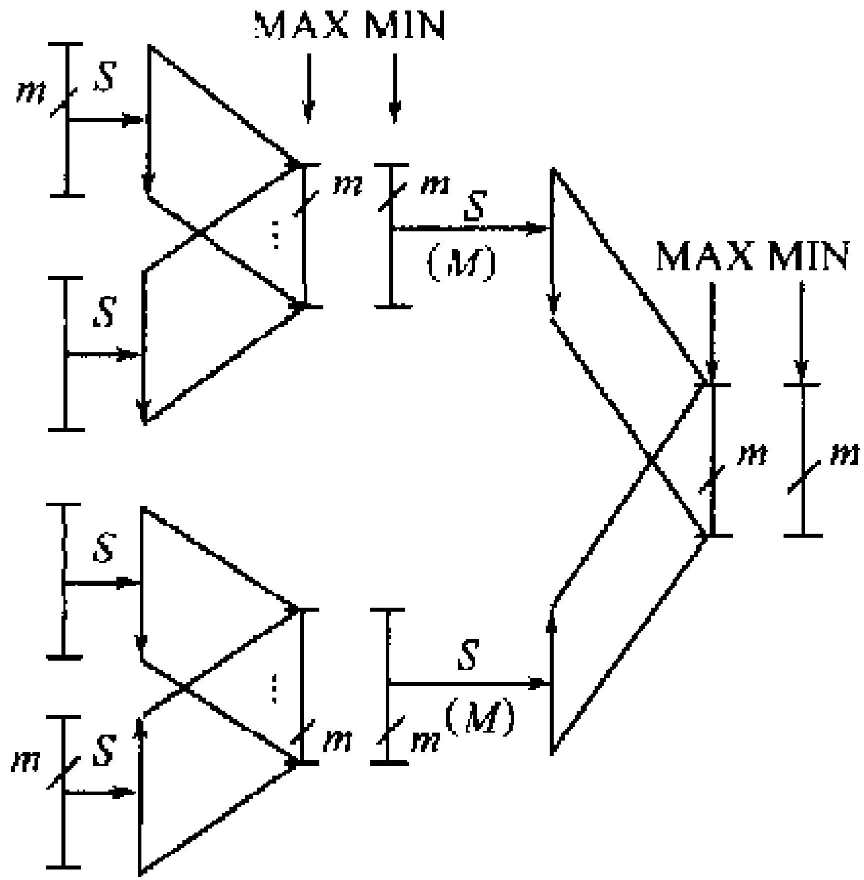
②使用Batcher排序网络对各子序列排序；

③将有序子序列形成双调序列，进行两两对接；

使用Batcher定理形成MAX,MIN序列，弃去MAX序列；

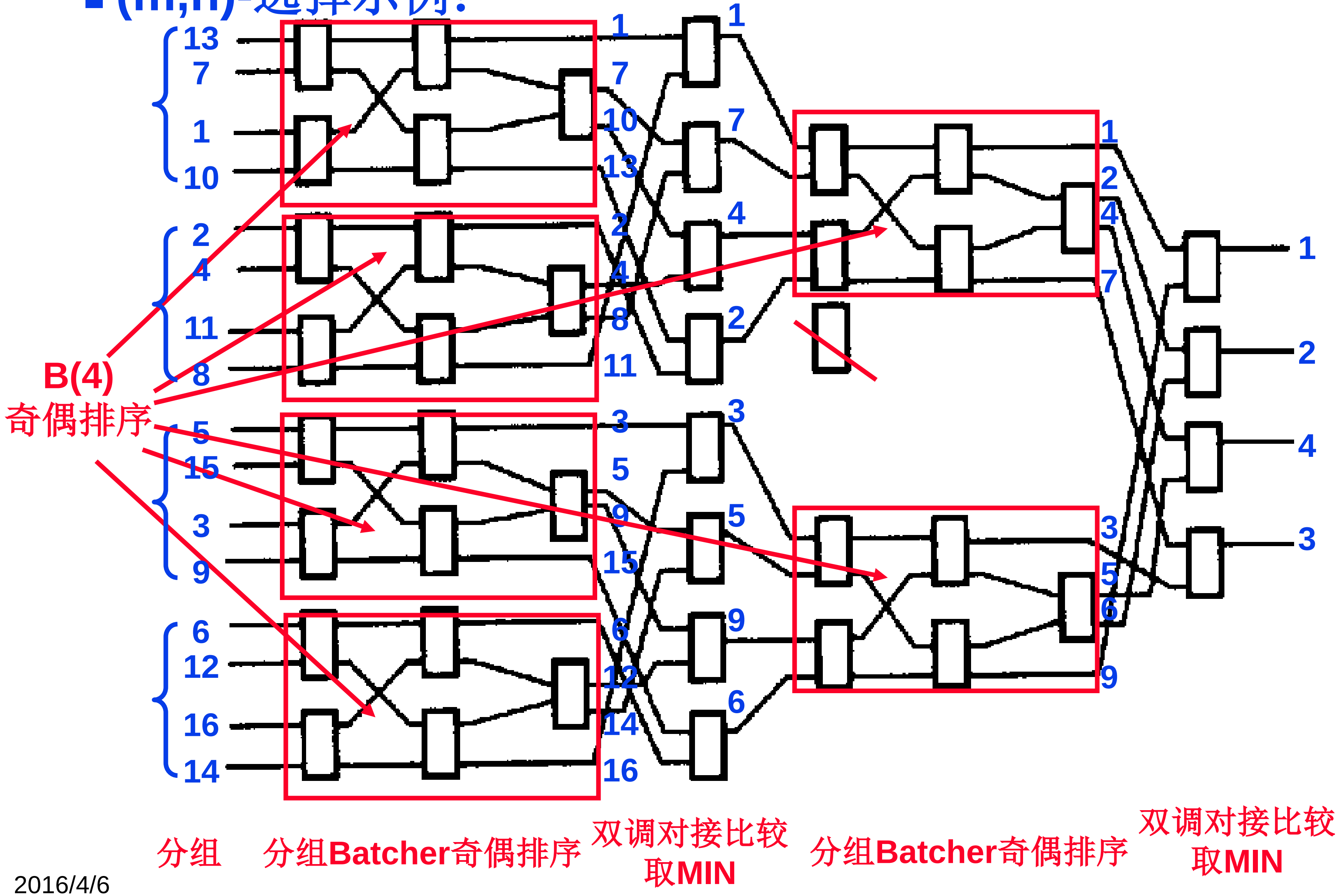
再使用Batcher排序网络将MIN序列排成有序序列；

④重复③直至MIN序列恰好包含所需的m个最小元素为止。

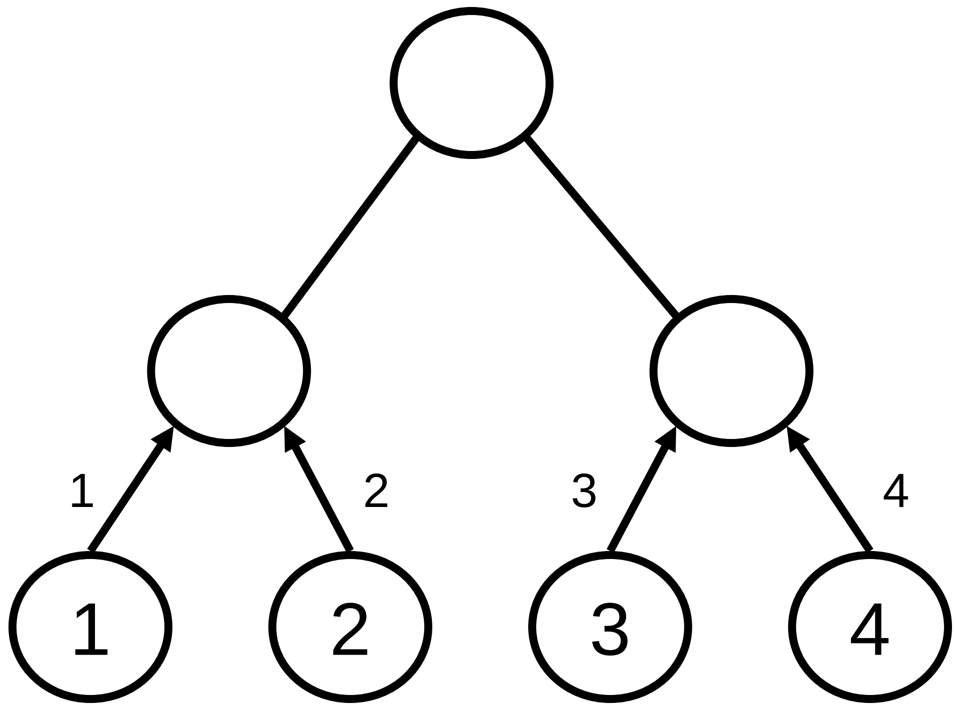


7.1.4 功能划分

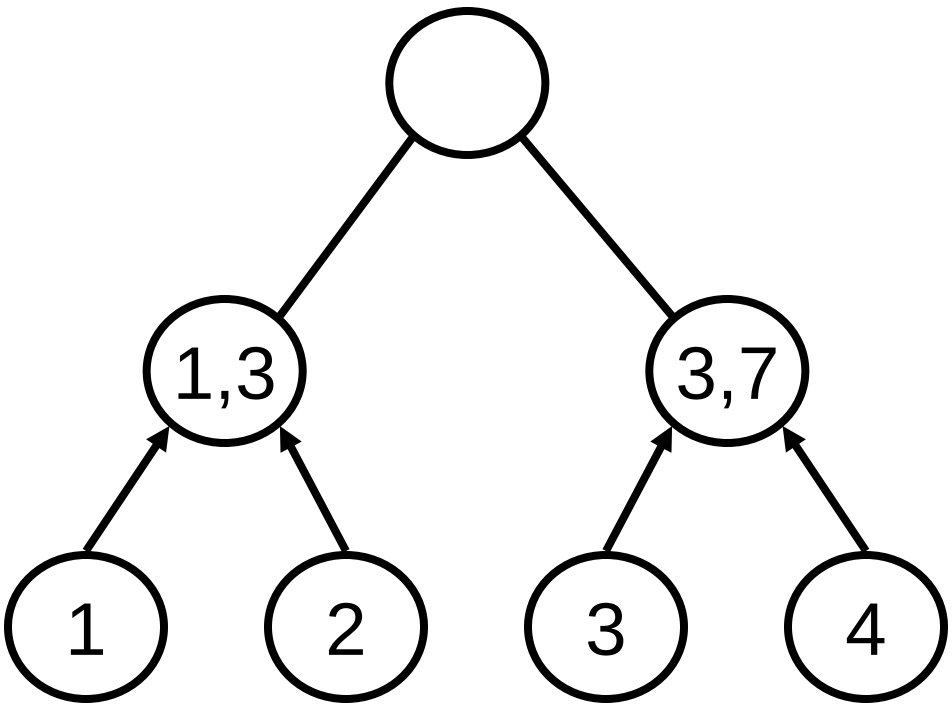
- (m,n)-选择示例：



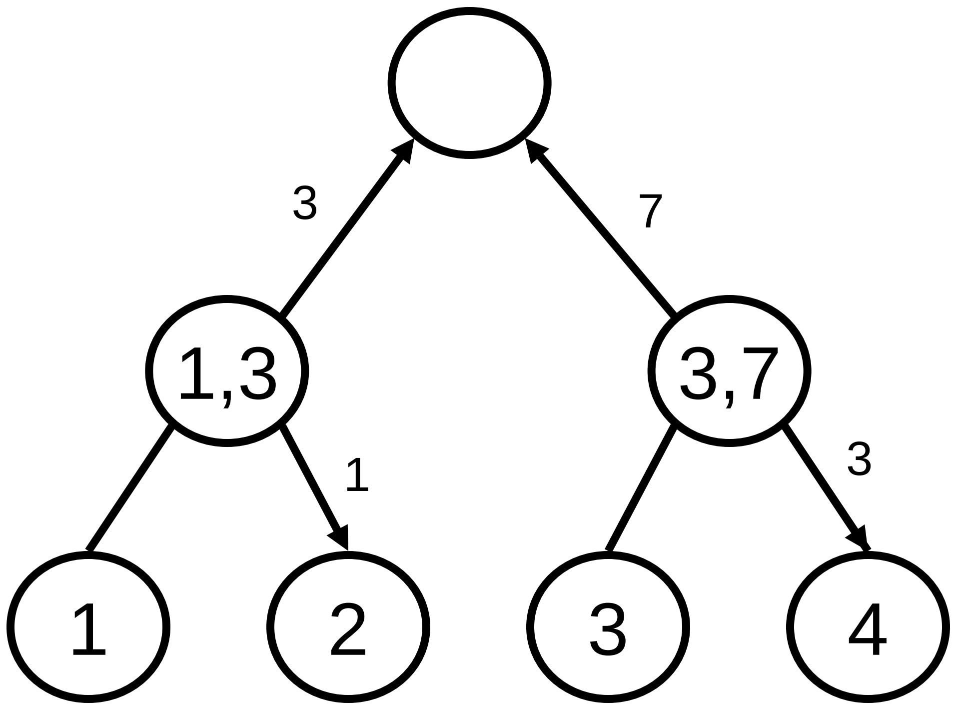
Prefix Sums on a Tree: More



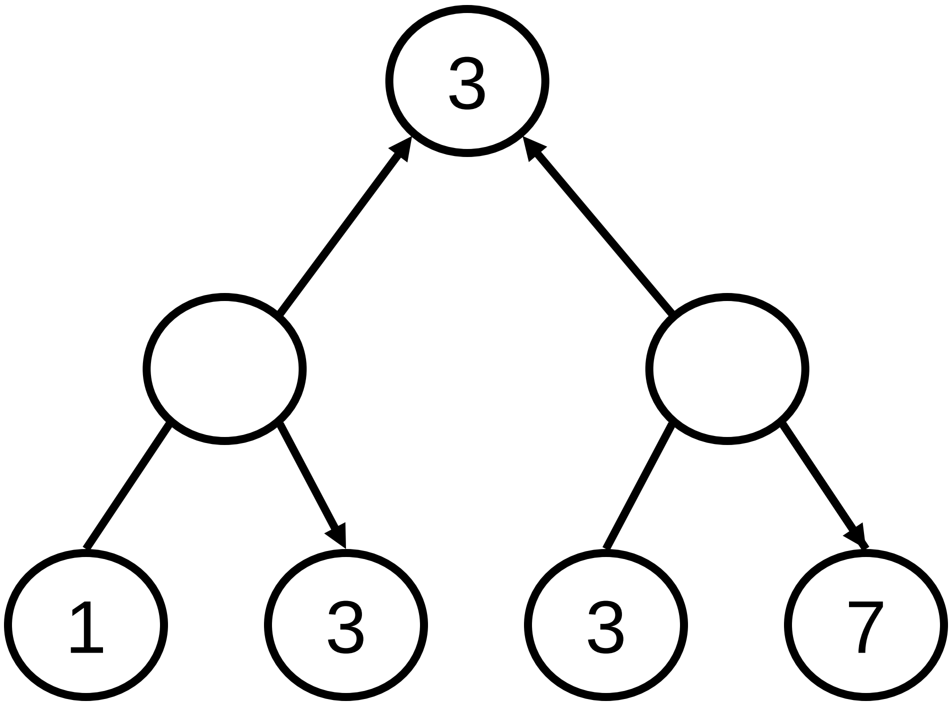
Prefix Sums on a Tree: More



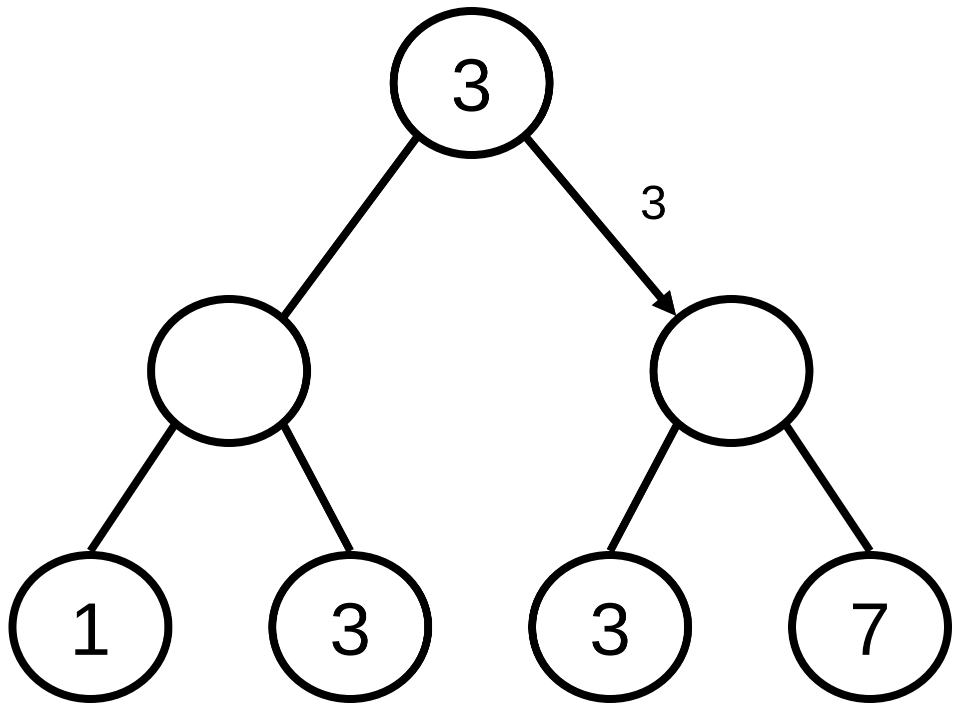
Prefix Sums on a Tree: More



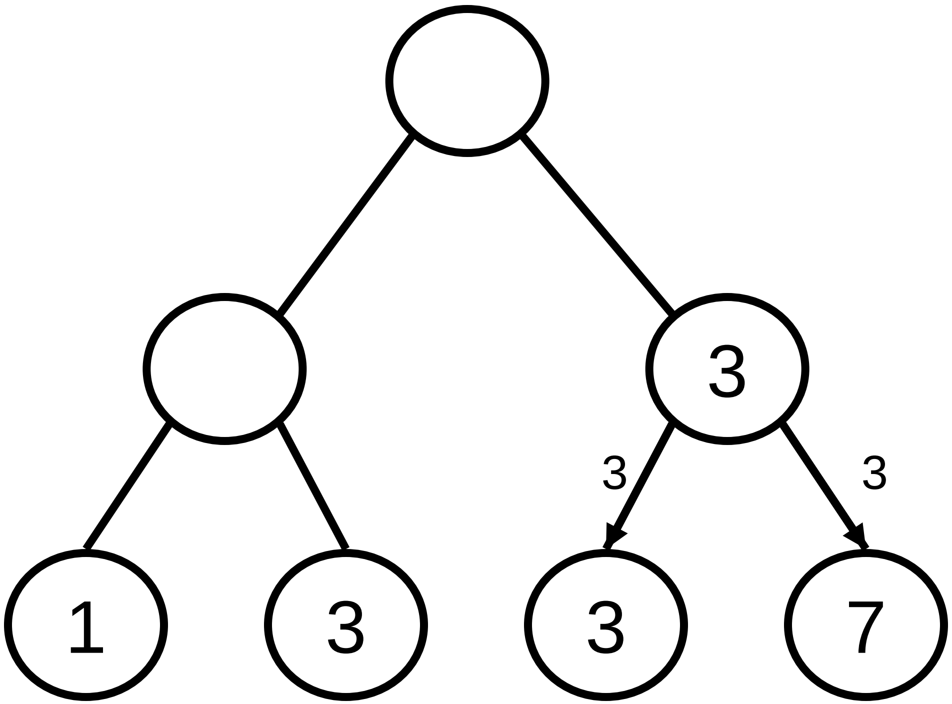
Prefix Sums on a Tree: More



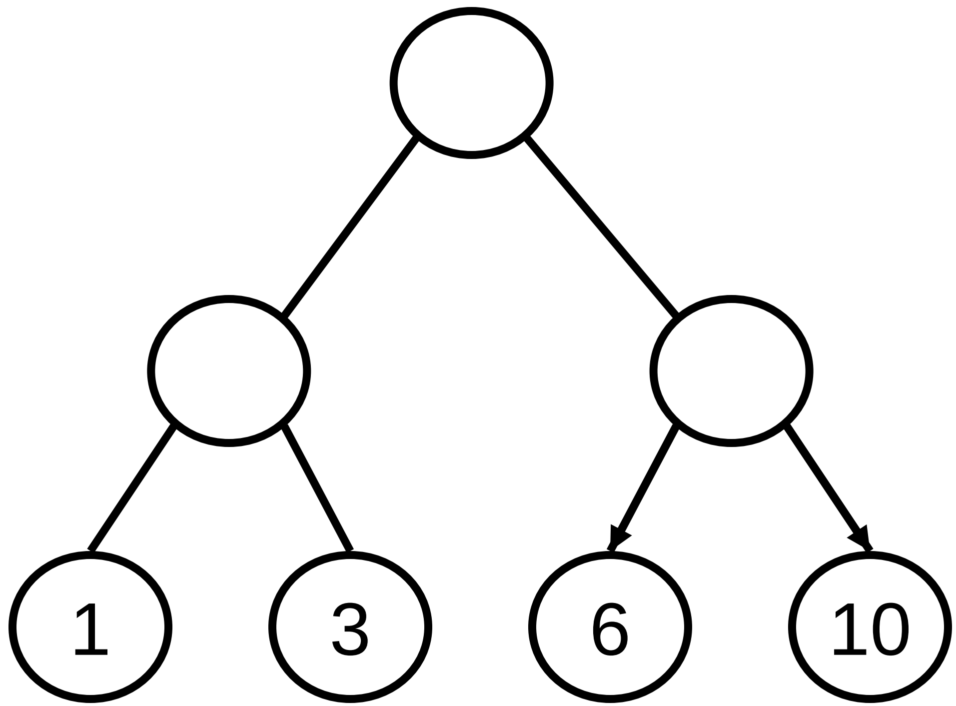
Prefix Sums on a Tree: More



Prefix Sums on a Tree: More



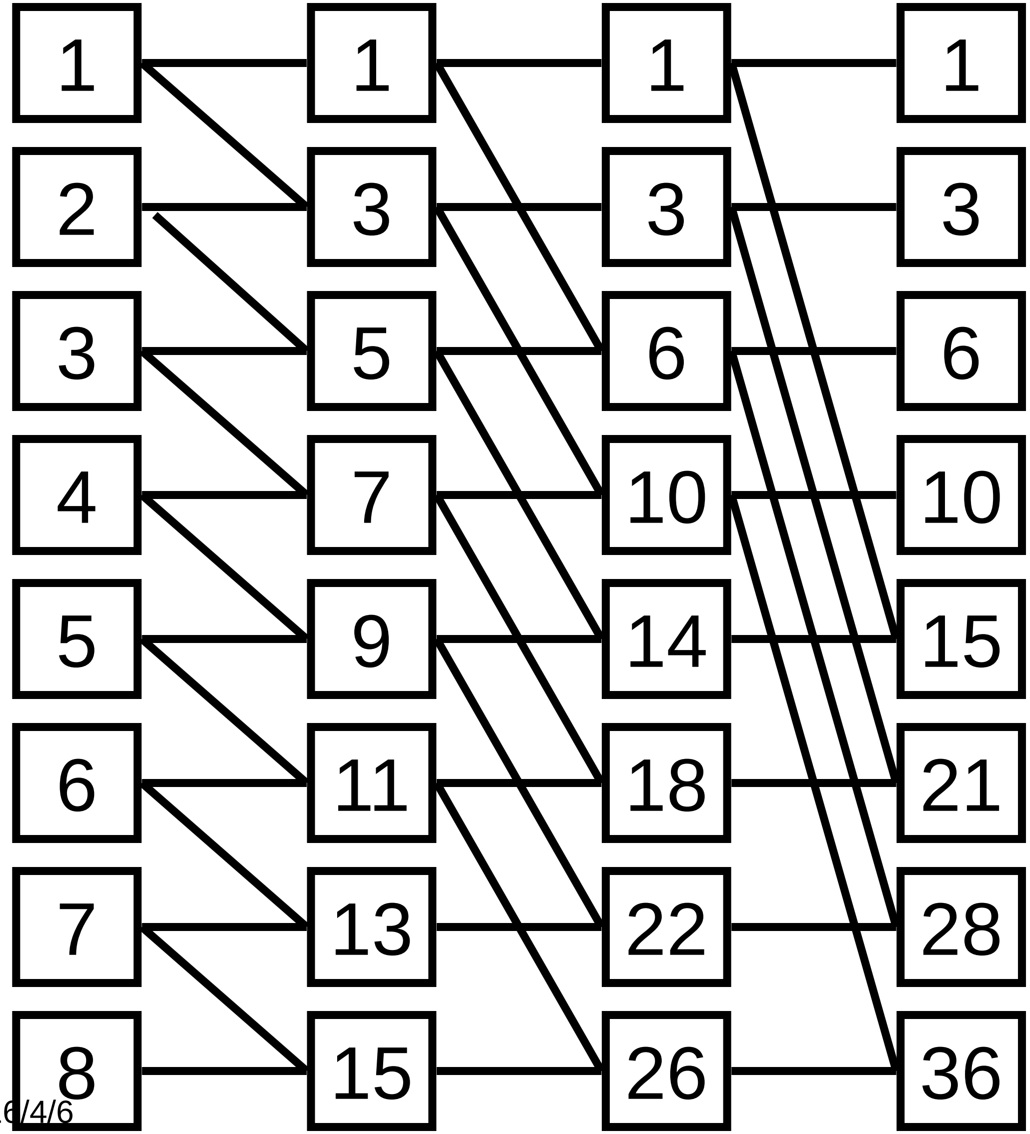
Prefix Sums on a Tree: More



Prefix Sums on a Tree: More

- Properties:
  - **time:** **2 log n**
  - **processors:** **$2n - 1$**
  - **cost** **$O(n \log n)$**
- Comparison with PRAM algorithm
  - **asymptotically equivalent**
  - **in practice, less efficient**
    - **weaker model**

7.3 平衡树设计技术



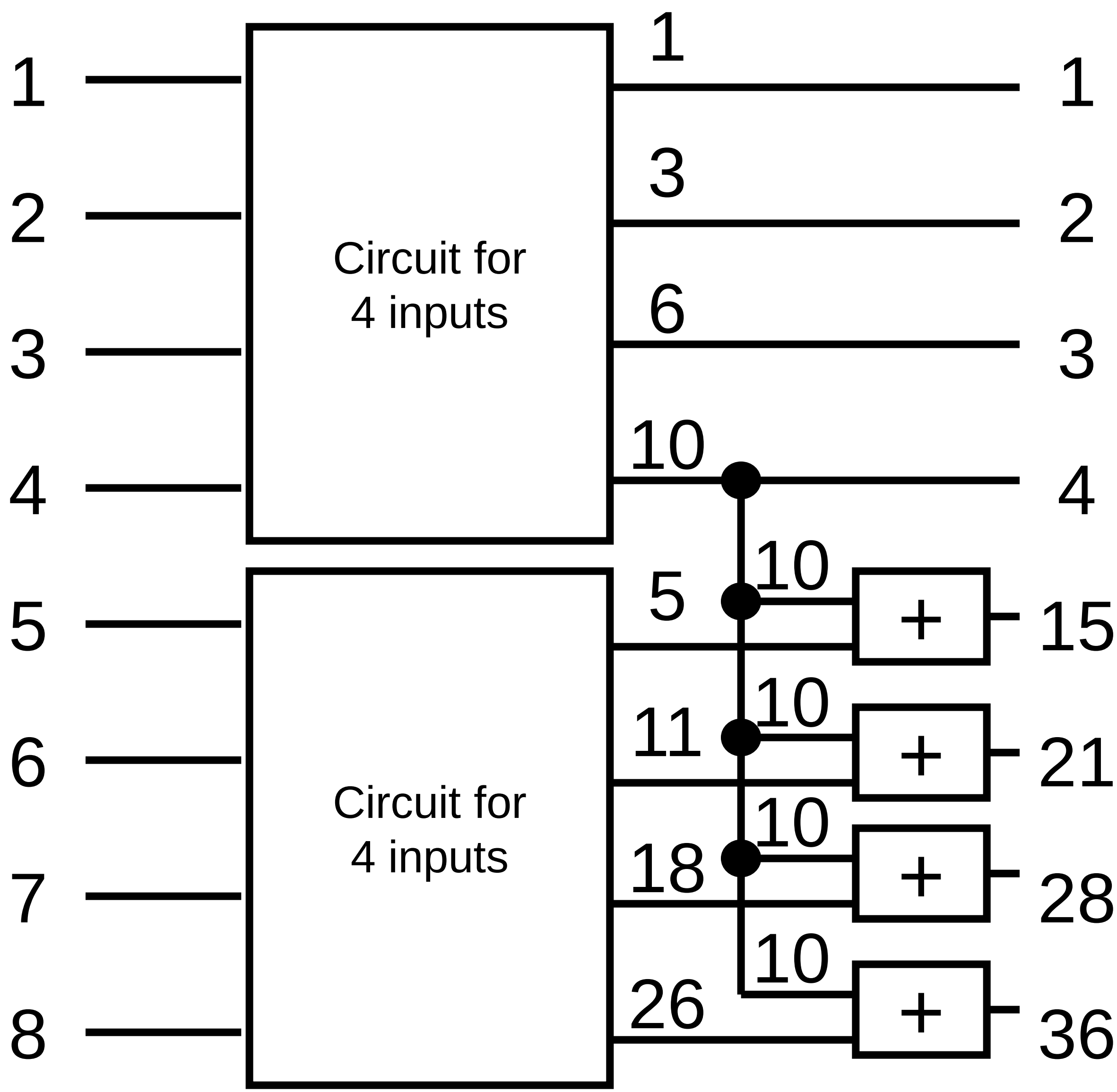
depth (time) = 3

complexity (cost) $= 3 \times 8$

$= 24$

Specialized Circuit

7.3 平衡树设计技术

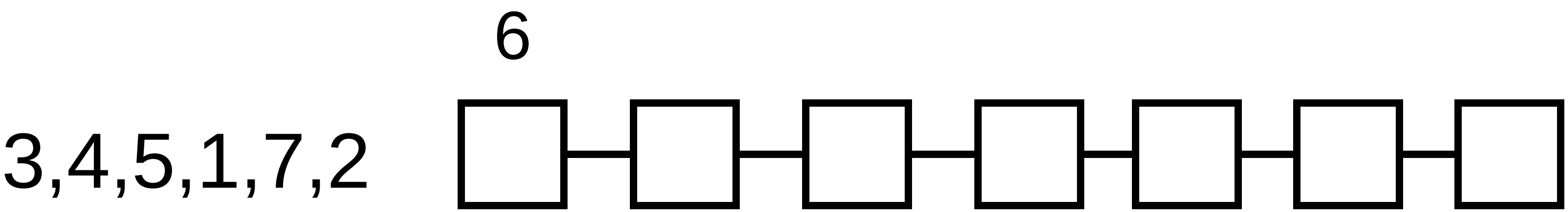


Recursive Circuit

7.5 流水线设计技术

Systolic排序算法示例: Sorting 3, 4, 5, 1, 7, 2, 6

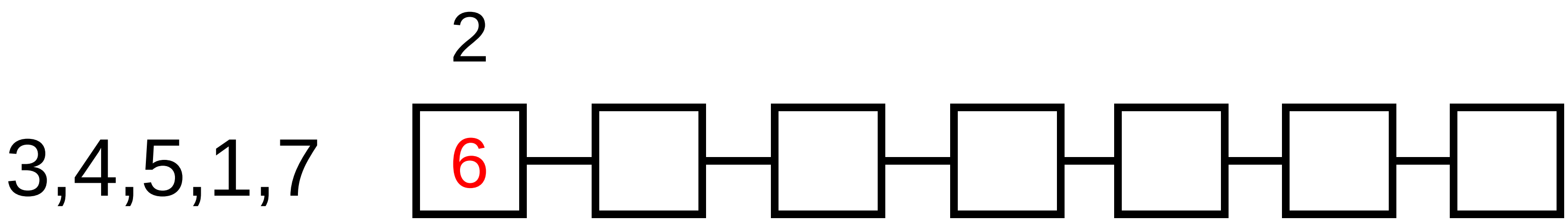
Step 1



7.5 流水线设计技术

Systolic排序算法示例: Sorting 3, 4, 5, 1, 7, 2, 6

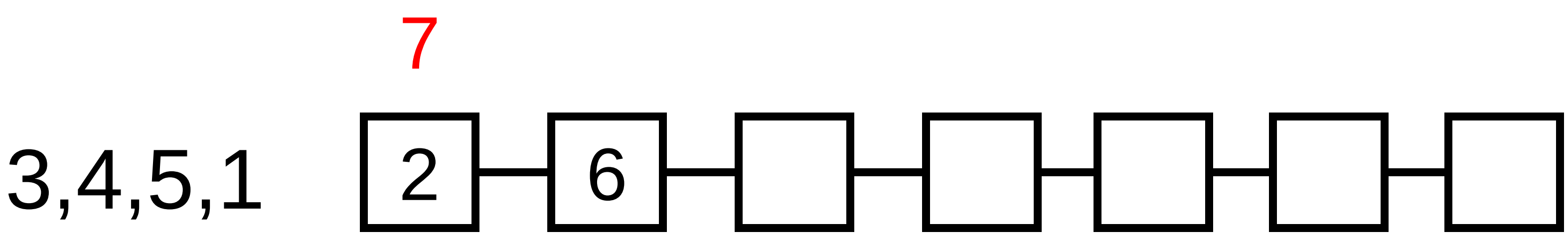
Step 2



7.5 流水线设计技术

Systolic排序算法示例: Sorting 3, 4, 5, 1, 7, 2, 6

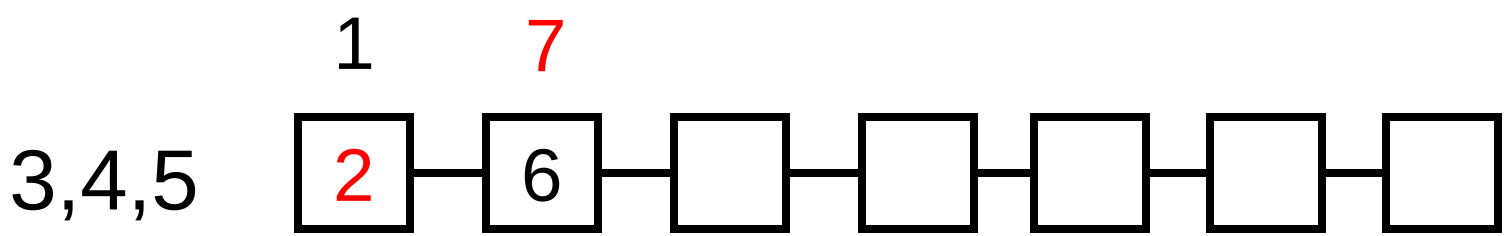
Step 3



7.5 流水线设计技术

Systolic排序算法示例: Sorting 3, 4, 5, 1, 7, 2, 6

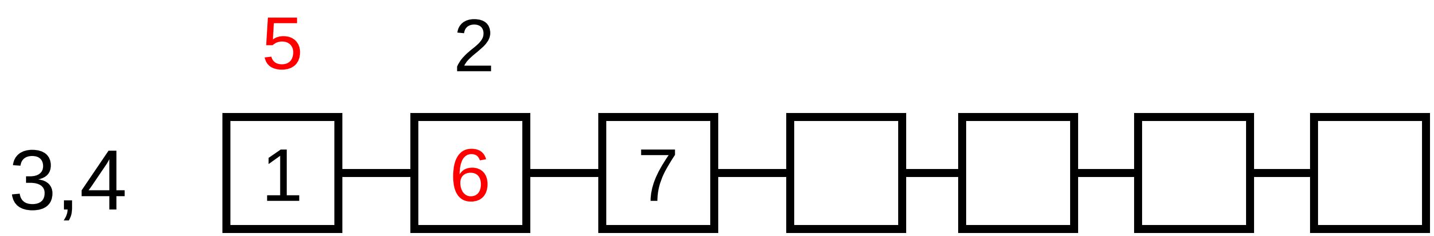
Step 4



7.5 流水线设计技术

Systolic排序算法示例: Sorting 3, 4, 5, 1, 7, 2, 6

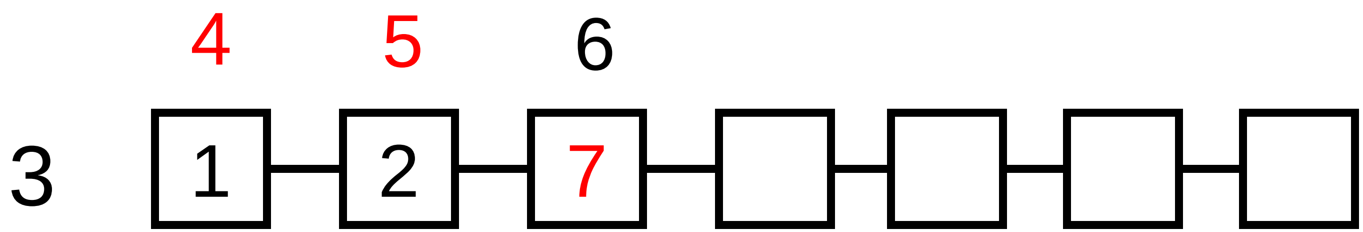
Step 5



7.5 流水线设计技术

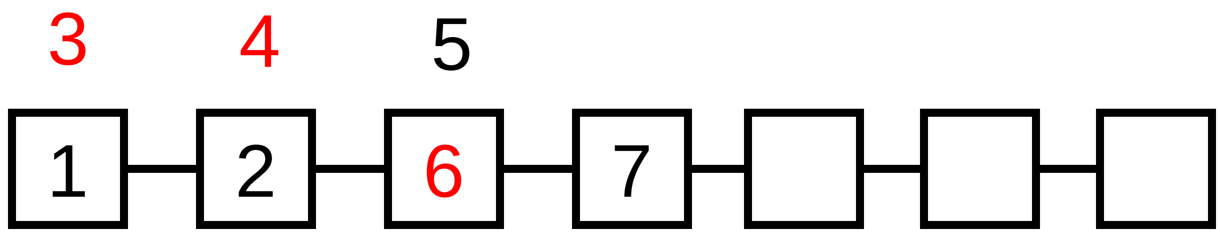
Systolic排序算法示例: Sorting 3, 4, 5, 1, 7, 2, 6

Step 6



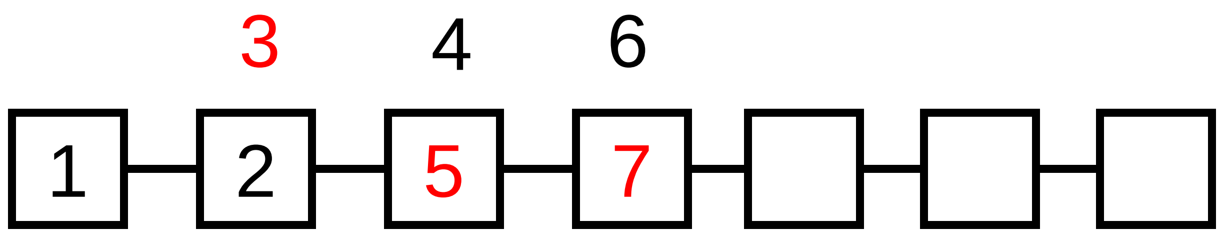
Systolic排序算法示例: Sorting 3, 4, 5, 1, 7, 2, 6

Step 7



Systolic排序算法示例: Sorting 3, 4, 5, 1, 7, 2, 6

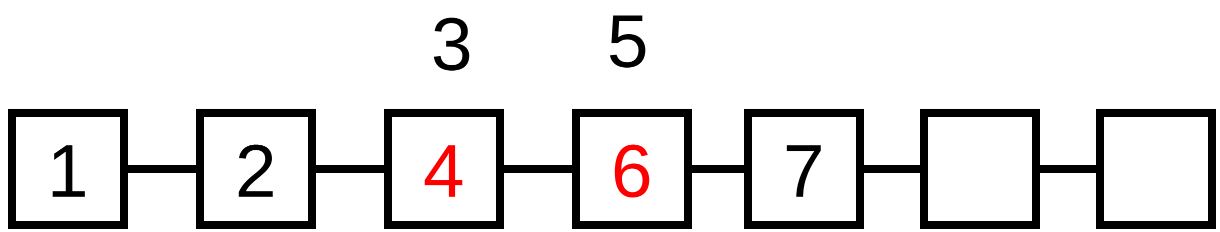
Step 8



7.5 流水线设计技术

Systolic排序算法示例: Sorting 3, 4, 5, 1, 7, 2, 6

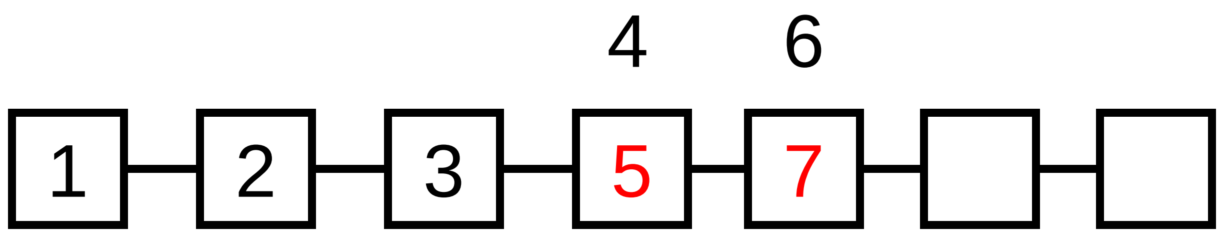
Step 9



7.5 流水线设计技术

Systolic排序算法示例: Sorting 3, 4, 5, 1, 7, 2, 6

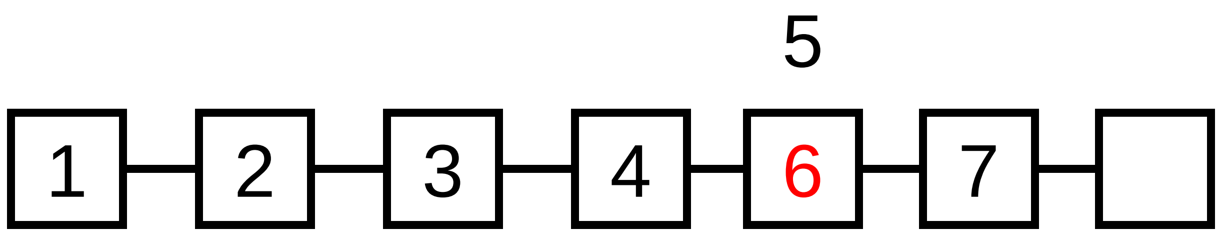
Step 10



7.5 流水线设计技术

Systolic排序算法示例: Sorting 3, 4, 5, 1, 7, 2, 6

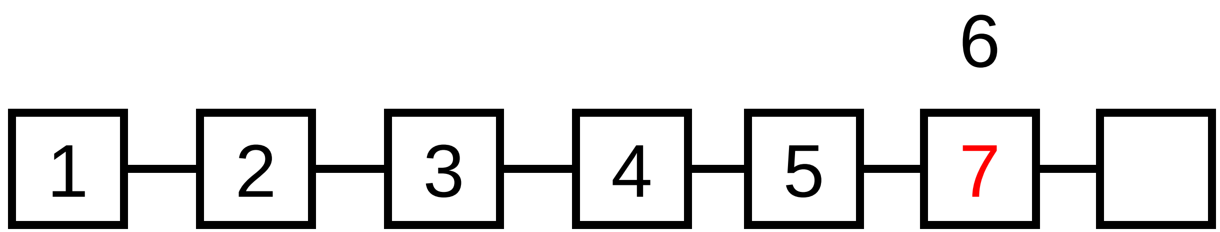
Step 11



7.5 流水线设计技术

Systolic排序算法示例: Sorting 3, 4, 5, 1, 7, 2, 6

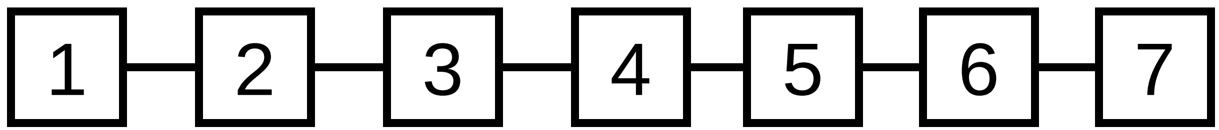
Step 12



7.5 流水线设计技术

Systolic排序算法示例: Sorting 3, 4, 5, 1, 7, 2, 6

Step 13



Activity 2

1.以下是上三角方程组回代解法的串行算法的形式化描述。（算法10.1）

输入： A_{n*n} $b = (b_1, \ldots, b_n)^T$ 输出： $x = (x_1, \ldots, x_n)^T$

```
Begin
   (1)for i=n downto 1 do
      (1.1)x_i=b_i/a_ii
      (1.2)for j=1 to i-1 do
              b_j=b_j-a_ji x_i
              a_ji=0
           endfor
      endfor
End
```

①请指出串行算法哪些部分可以并行化。②写出并行算法的形式化描述（需要注明计算模型类型），分析你的算法的时间复杂度。

2. 习题7-10（P207）