拓展1：基于伸展树的区间操作

(1) 区间提取

给定一个数列$[n_1, n_2, \cdots, n_m]$，假设我们需要提取区间$[n_a, n_b]$的子序列。

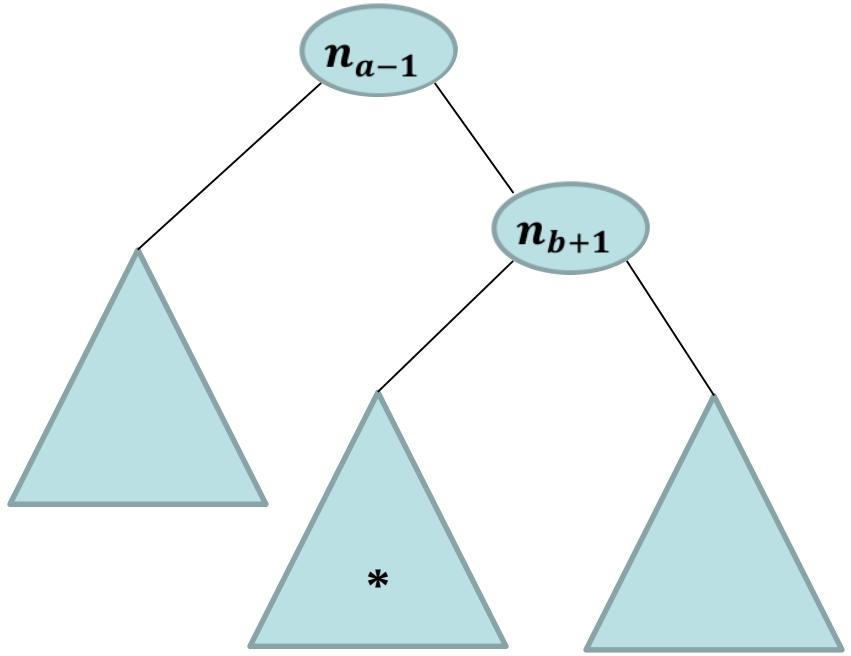
步骤：

Step 1：将位置n_a前面的元素转到树根；

Step 2：将位置n_b后面一个元素转到树根的右儿子；

Step 3：则树根右儿子的左子树就是需要提取的区间。

基于伸展树的区间操作



$[n_a, n_b\ ]$

例如： A=[1,2,3,4,5,6,7,8,9,10]
提取 [4---8]

基于伸展树的区间操作

(2) 区间删除

给定一个数列$[n_1, n_2, \cdots, n_m]$，假设我们删除区间$[n_a, n_b]$子序列。

步骤：

Step 1：利用区间提取方法找到区间所对应的子树；

Step 2：删除对应的子树

基于伸展树的区间操作

(3) 区间插入

给定一个数列A=[n_1, n_2, …, n_m]和对应的伸展树，假设我们在n_a和n_{a+1}之间插入一个新序列B。

Step 1：把n_a转到树根；

Step 2：把n_{a+1}转到树根的右儿子；

Step 3：将待插入序列B 构建成一颗伸展树。

Step 4：把新构建的伸展树插入到树根的右儿子的左子树。

基于伸展树的区间操作

(4) 区间翻转

给定一个数列A=[n_1，n_2，…，n_m]和对应的伸展树，假设我们在n_a和n_b之间插入一个子序列做一个前后翻转。

Step 1：按照区间提取算法，定位[n_a，n_b]所对应的子树；

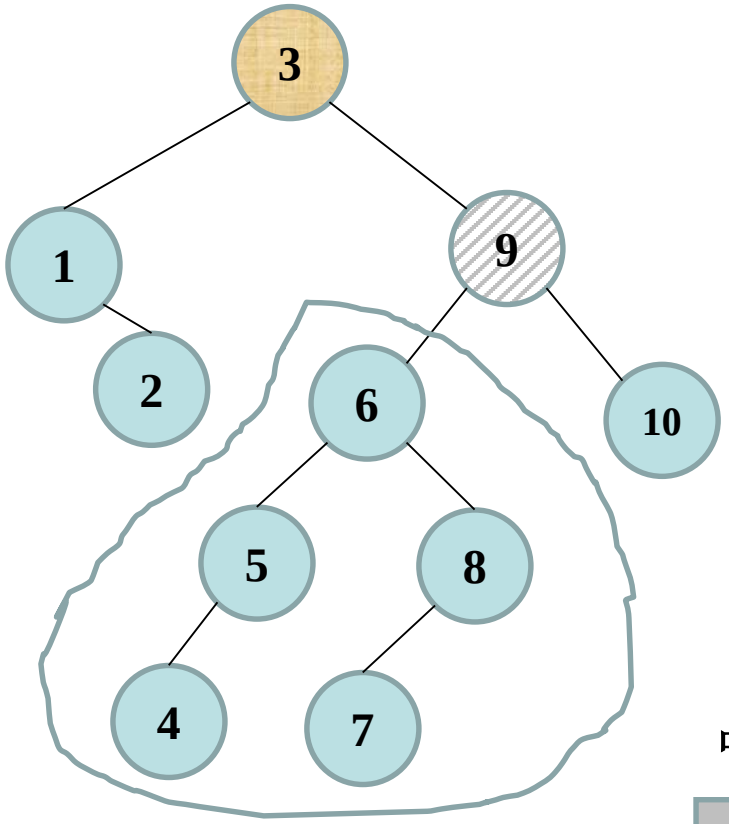
Step 2：对此区间中的每个结点，交换它的左右儿子。

基于伸展树的区间操作

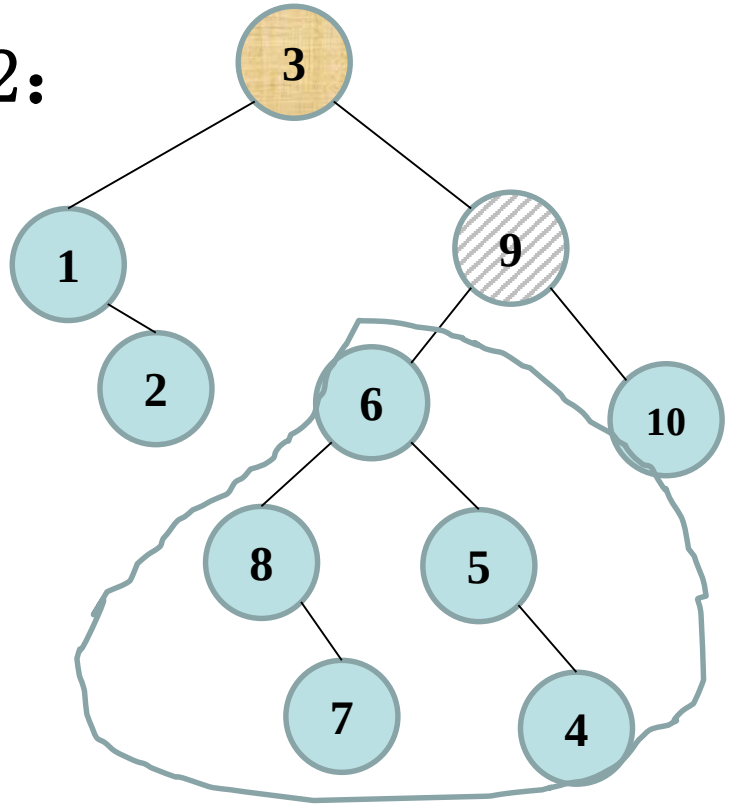
(4) 区间翻转

例如： **A=[1,2,3,4,5,6,7,8,9,10], 翻转A[4-8]部分。**

Step 1：



Step 2：



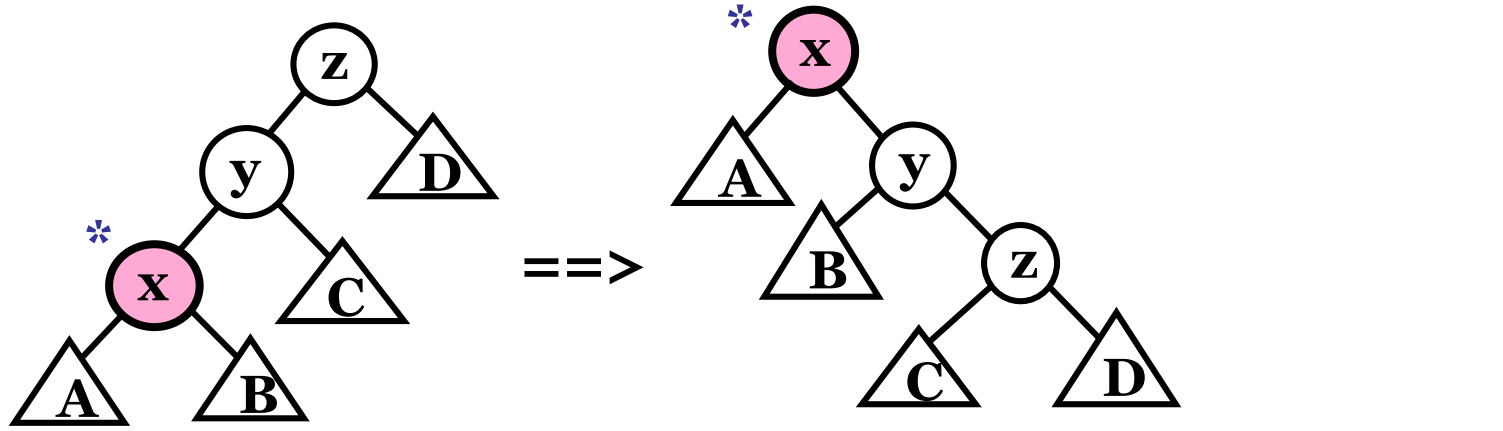
中序访问

A=[1, 2, 3, 8, 7, 6, 5, 4, 9, 10],

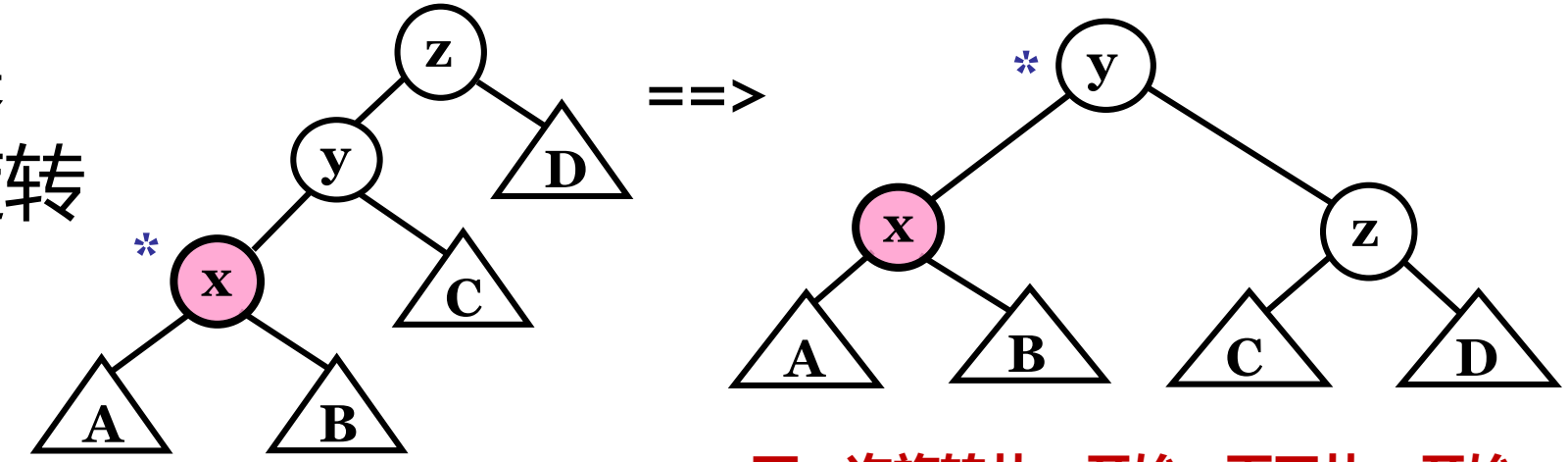
拓展2：半伸展树

下一轮的伸展，当前结点上移到父结点，父结点位置开始旋转，即半伸展不需要把检索结点移动到根

普通
一字旋转

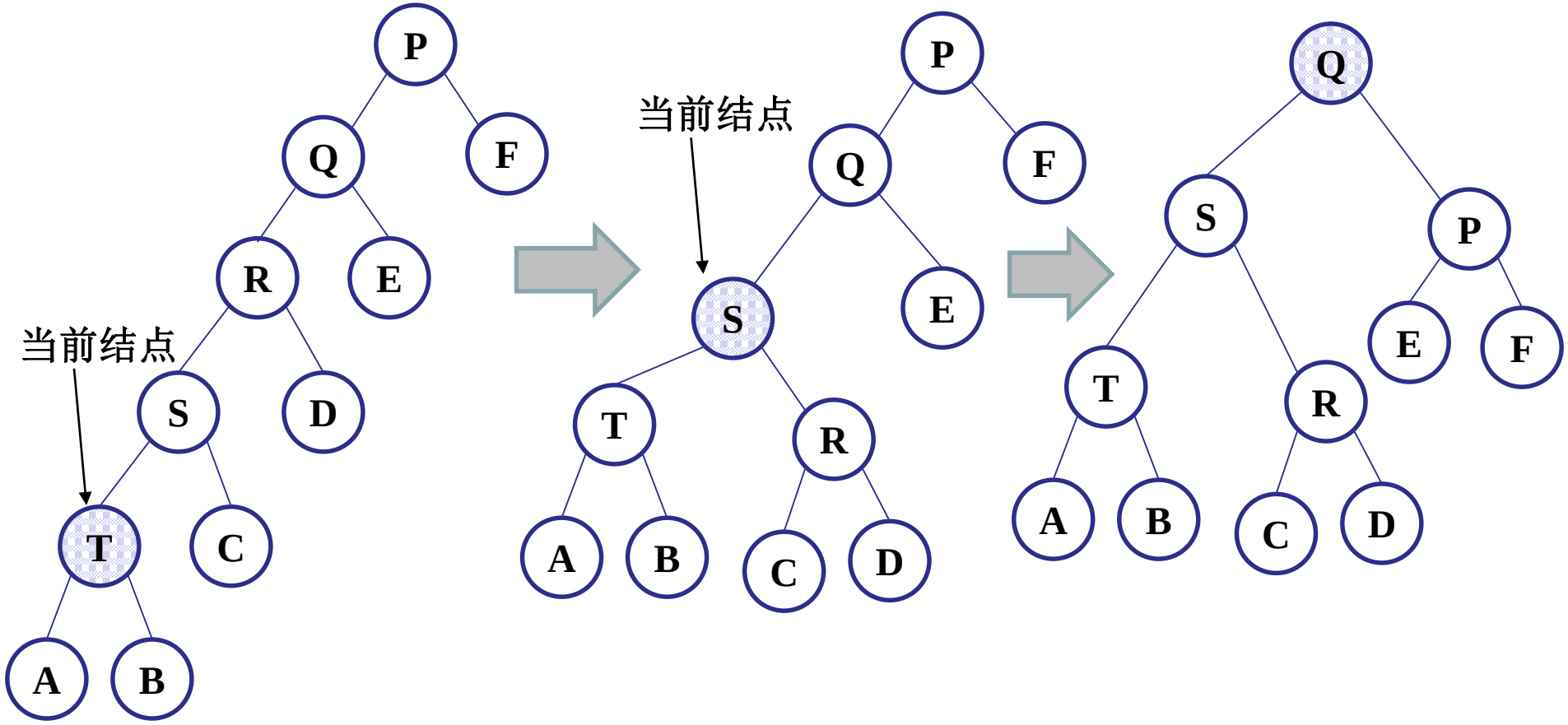


半伸展
一字旋转



下一次旋转从 y 开始，而不从 x 开始

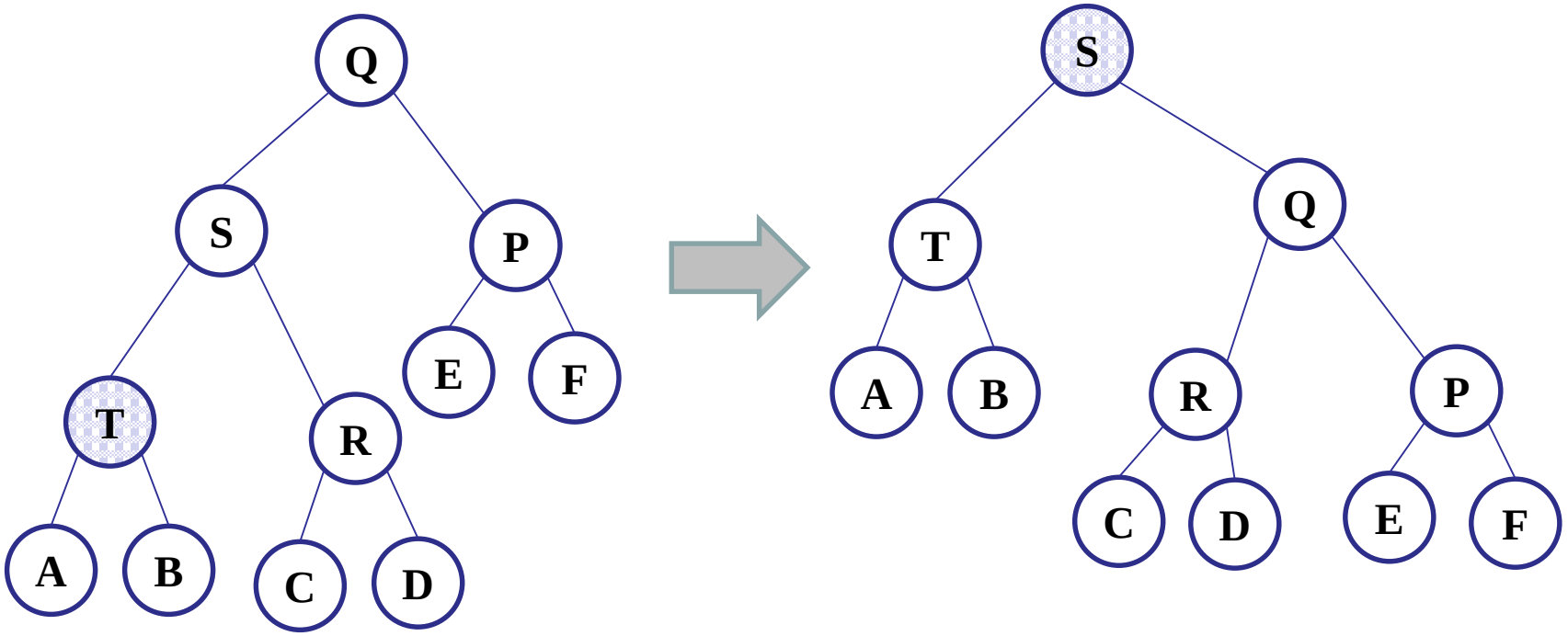
下一轮的伸展，当前结点上移到父结点，父结点位置开始旋转，即半伸展不需要把检索结点移动到根



第一次访问T

半伸展树

下一轮的伸展，当前结点上移到父结点，父结点位置开始旋转，即半伸展不需要把检索结点移动到根



第二次访问T