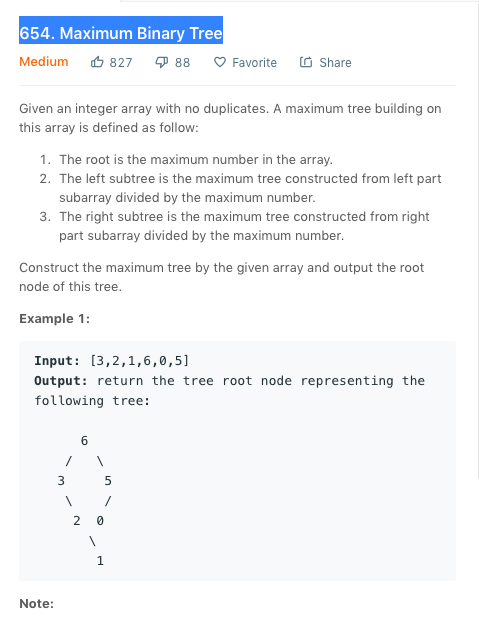
**654. Maximum Binary Tree**

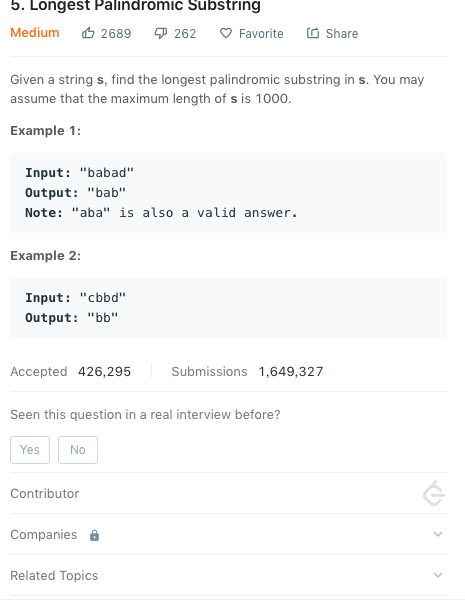


思路：一个递归的简单练习

f( A, i, j)

用循环找到 A 数组中 i-j 之间的最大元素，设该元素的下标为k ,用A[k] new 出一个节点x，将A[I,k-1] 和A[k+1,j]递归，将 x.left 设为 f( A , i, k-1) ,x.right 设为 f(A ,K+1,j),并返回x 节点

**5. Longest Palindromic Substring**



这题的做法是动态规划，长度为k 的回文串可以从长度为k-2 的回文串产生

比如说 A[i,j] 的长度 等于 A[i+1,j-1] +2 ，假如 A[i+1,j-1]是回文串切 s[i] = s[j]

所以先把小的回文串处理了，可以很自然地处理大的回文串

接下来是动态规划的做法：

1. 首先预处理长度为1，和长度为2 的A[i,j]
2. 接下来是循环过程：

For k = 3 to n

for i = 1 to n – k

j = i+k

if ( A[i + 1,j -1] != -1 && s[i] == s[j])

A[i,j] = A[i+1,j-1]+2

Else

A[i,j] = -1

以上过程应该顺便把最长的长度和相应的下标记录，