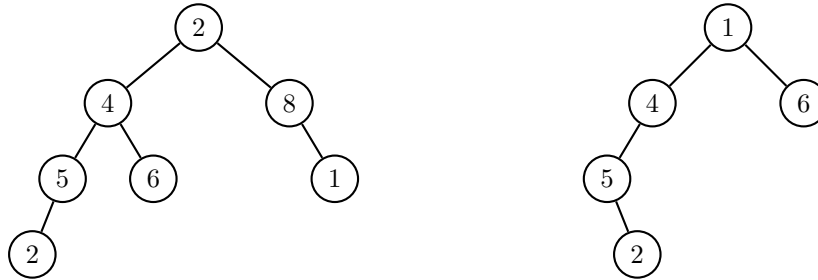


# Camino de pares

Definimos un *camino* entre dos nodos de un árbol binario como una secuencia de nodos  $n_1 n_2 \dots n_k$  sin repeticiones (por cada nodo del árbol se pasa como mucho una vez) tal que para todo par de nodos consecutivos  $n_i n_{i+1}$  ( $1 \leq i < k$ ) uno de ellos siempre es padre del otro ( $n_i$  es padre de  $n_{i+1}$  o  $n_{i+1}$  es padre de  $n_i$ ). Definimos la *longitud* de un camino  $n_1 n_2 \dots n_k$  como el número de nodos que lo forman,  $k$ .

Dado un árbol binario de números enteros positivos queremos calcular la longitud del camino más largo formado todo él por números pares. Por ejemplo, el siguiente árbol de la izquierda tiene el camino 6 4 2 8 de longitud 4. En cambio, en el árbol de la derecha todos los caminos formados únicamente por números pares tienen longitud 1.



*Requisitos de implementación.*

Se implementará una función externa a la clase `bintree` que explore el árbol calculando la longitud del camino más largo con solo números pares.

## Entrada

La entrada comienza indicando el número de casos de prueba que vendrán a continuación. Cada caso consiste en una serie de números enteros positivos con la descripción de un árbol binario: el árbol vacío se representa con el valor -1; un árbol no vacío se representa con el valor de un nodo (que denota la raíz), seguido primero de la descripción del hijo izquierdo y después de la descripción del hijo derecho.

## Salida

Para cada caso, se escribirá la longitud del camino más largo entre dos nodos del árbol, formado todo él por números pares.

## Entrada de ejemplo

```
3
2 4 5 2 -1 -1 -1 6 -1 -1 8 -1 1 -1 -1
1 4 5 -1 2 -1 -1 -1 6 -1 -1
3 4 2 -1 -1 6 -1 8 -1 -1 5 4 -1 2 -1 -1
```

## Salida de ejemplo

```
4
1
4
```

**Autor:** Isabel Pita.