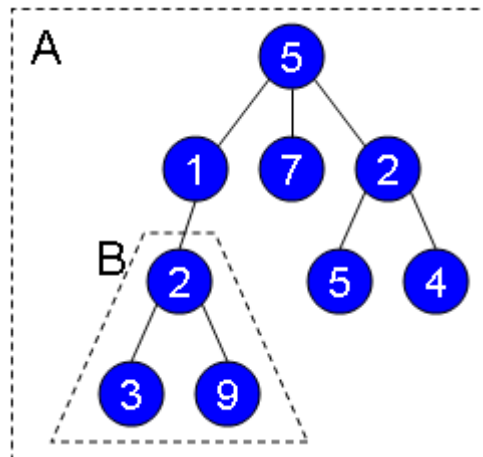
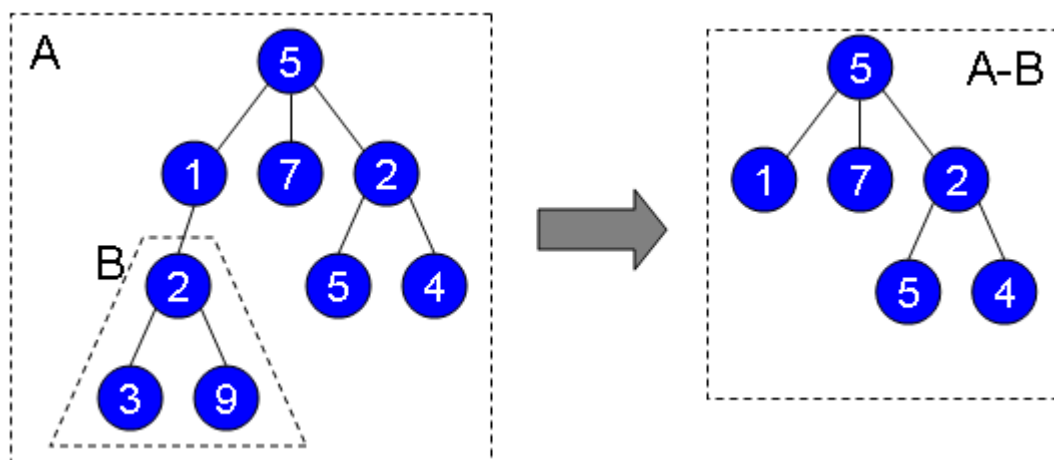


“Mitad” de un árbol

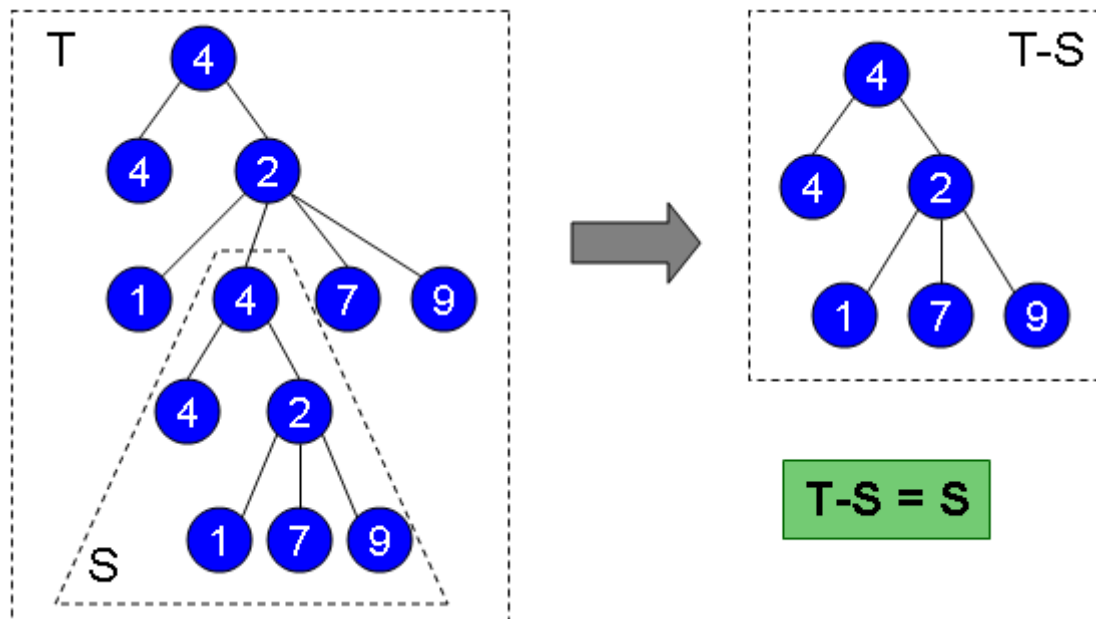
Dentro de un árbol, un subárbol está compuesto por un nodo diferente de la raíz y todos sus descendientes. En la siguiente figura podrá observar un árbol **A** que contiene al subárbol **B**.



A un árbol se le puede eliminar todo un subárbol. Si al árbol **A** de la figura anterior se le elimina el subárbol **B**, se obtiene el árbol “**A-B**” siguiente:



Es posible que al eliminar un subárbol **S** de un árbol **T**, el árbol resultante sea igual al subárbol que se eliminó, como se muestra a continuación:



En ese caso, decimos que el subárbol **S** es la “mitad” del árbol **T**. Note que esto no quiere decir que todo árbol tenga que tener mitad. Por ejemplo el árbol **A** del primer ejemplo, no tiene mitad.

Dentro de la plantilla de solución que se le suministra, implemente el método

```
public static Arbol<T> Mitad<T>(Arbol<T> a)
```

que dado un árbol debe devolver un subárbol “mitad” del árbol que se le pasa como parámetro. En caso de no existir tal mitad, entonces debe devolver **null**.

Aclaraciones

- Tras la ejecución del método **Mitad**, el árbol que se le pasó como parámetro debe estar en idénticas condiciones que antes de invocar el método, sin que le sobre o falte ninguno de sus nodos originales.
- Se garantiza que el árbol pasado como parámetro nunca será **null** y que ninguno de sus nodos tendrá valor **null**.
- El proyecto **Prueba** que se encuentra dentro de la plantilla de solución no será evaluado. En él puede hacer lo que desee para probar su solución.
- La clase **Arbol<T>**, similar a la utilizada en las clases de programación, ya viene definida en una biblioteca de clases que se ha incluido como referencia dentro de su solución. En el proyecto **Prueba** encontrará código de ejemplo que le ilustrará cómo crear instancias de esta clase.