

23 NSI 25

```
In [ ]: class Arbre:
        def __init__(self, etiquette):
            self.v = etiquette
            self.fg = None
            self.fd = None
```

```
In [ ]: def parcours(arbre, liste):
        if arbre != None:
            parcours(arbre.fg, liste)
            liste.append(arbre.v)
            parcours(arbre.fd, liste)
        return liste
```

```
In [ ]: def insere(arbre, cle):
        """ arbre est une instance de la classe Arbre qui implémente
            un arbre binaire de recherche.
        """
        if ...:
            if ...:
                insere(arbre.fg, cle)
            else:
                arbre.fg = Arbre(cle)
        else:
            if ...:
                insere(arbre.fd, cle)
            else:
                arbre.fd = Arbre(cle)
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js