Exercise 1 - Design and Documentation

1. If the `source` and `target` change often.
2. The documentation should include what happens when there is no shortest path.
3. (a) Save the result of `shortestPath` to an object property. Reset to `null` when `addEdge` or `setST` is called.

   (b) Could maybe indicate that repeated calling of `shortestPath` only calculates the result once unless the modifying methods are called.

   (c) Postcondition to change is that the stored path is `null`. Postcondition for `stortestPath` is that the stored path should be the methods result.

   (d)

Exercise 2 - Design

1. ```
   Integer[] arr = {1, 2, 3};
   List<Integer> a = List(arr, 3);
   a.take();
   a.set(2, 0);
   ```

   or

   ```
   hi there
   ```

2. ```
   class List<E>{
       E[] elems;
       int len;
       boolean shared;

       List(E[] e, int l) {
           elems = e;
           len = l;
           shared = true;
       }

       void set(int index, E e) {
           if(shared) {
               elems = elems.clone();
               shared = false;
           }
           elems[index] = e;
       }
   ```

```
    List<E> take(){
        shared = true;
        return new List<E>(elems, len - 1);
    }
}
```