```java
/**
 * Helper method for recursive selection sort.
 * Sorts the first n elements of the list.
 * In each step, it finds the maximum element in the unsorted part and places it at the end of that par
 * @param list The list of PlayerSpells to sort.
 * @param n The number of elements from the beginning of the list to consider for sorting.
 */
private void selectionSortInventory(List<PlayerSpell> list, int n) {
    // base case
    if (n <= 1) {
        return;
    }

    // Find the index of the maximum element in the subarray list
    int indexOfMax = 0;
```

```
    for (int i = 1; i < n; i++) {
        if (list.get(i).finalAPCost() > list.get(indexOfMax).finalAPCost()) {
            indexOfMax = i;
        }
    }

    // Swap the maximum element with the element at the current end of the unsorted portion (n-1)
    if (indexOfMax != n - 1) {
        PlayerSpell temp = list.get(indexOfMax);
        list.set(indexOfMax, list.get(n - 1));
        list.set(n - 1, temp);
    }

    // Recursively call for the remaining n-1 elements (excluding the now sorted largest element)
    selectionSortInventory(list, n - 1);
}
```