**Frequency Cost Operations**

**int** rank (**int**[] collection, **int** target) {

**int** lo = 0; \_1\_ \_\_ \_\_

**int** hi = collection.length-1; \_1\_ \_\_ \_\_

**while** (lo <= hi) { \_\_ \_\_ \_\_

**int** mid = (lo+hi)/2; \_\_ \_\_ \_\_

**int** rc = collection[mid] - target; \_\_ \_\_ \_\_

**if** (rc < 0) { \_\_ \_\_ \_\_

lo = mid+1; \_\_ \_\_ \_\_

} **else** **if** (rc > 0) { \_\_ \_\_ \_\_

hi = mid-1; \_\_ \_\_ \_\_

} **else** {

**return** mid; \_1\_ \_\_ \_\_

}

}

**return** -1; \_1\_ \_\_ \_\_

}

**int** rank (**int**[] collection, int lo, int hi, **int** target) {

**if** lo > hi { return -1; } \_1\_ \_\_ \_\_

**int** mid = (lo+hi)/2; \_1\_ \_\_ \_\_

**int** rc = collection[mid] - target; \_1\_ \_\_ \_\_

**if** (rc < 0) { \_\_ \_\_ \_\_

**return** rank(collection, mid+1, hi, target); \_\_ \_\_ \_\_

} **else** **if** (rc > 0) { \_\_ \_\_ \_\_

**return** rank(collection, lo, mid-1, target); \_\_ \_\_ \_\_

} **else** {

**return** mid; \_1\_ \_\_ \_\_

}

}