CSE 241 Algorithms and Data Structures **11/7/2008**

Quicksort and Partition

This handout gives the PARTITION algorithm we discussed in class, along with the QUICKSORT algorithm that uses it to sort an array. We give only the randomized form of the QUICKSORT algorithm, but the deterministic version can be derived by deleting the indicated lines.

Both algorithms take an array $A[p \ldots r]$. QUICKSORT sorts the array, while PARTITION rearranges it into two parts, the *low* and *high* parts, such that every element in the low part is strictly less than every element in the high part. Moreover, the *partition element*, which is initially $A[r]$, becomes the greatest element in the low part and is placed at its end. PARTITION returns the index of the partition element in the final array, which is also the end of the array's low part.

```
QUICKSORT(A, p, r)
    if (p < r) {
        x ← random(p, r) // randomized version only
        swap(A[x], A[r])    // randomized version only
        z ← PARTITION(A, p, r)
        QUICKSORT(A, p, z − 1)
        QUICKSORT(A, z + 1, r)
    }


PARTITION(A, p, r)
    i ← p − 1
    j ← p − 1
    do {
        j++
        if (A[j] ≤ A[r]) {
            i++
            swap(A[i], A[j])
        }
    }
    while (j < r − 1)

    swap(A[i + 1], A[r])
    return i + 1
```