

Worksheet #36: Parallelizing the Split step in Radix Sort

Name: _____

Netid: _____

The Radix Sort algorithm loops over the bits in the binary representation of the keys, starting at the lowest bit, and executes a split operation for each bit as shown below. The split operation packs the keys with a 0 in the corresponding bit to the bottom of a vector, and packs the keys with a 1 to the top of the same vector. It maintains the order within both groups. The sort works because each split operation sorts the keys with respect to the current bit and maintains the sorted order of all the lower bits. Your task is to show how the split operation can be performed in parallel using scan, reverse, not(Flags) operations, and to explain your answer.

```
1.A = [5 7 3 1 4 2 7 2]
2.A<0> = [1 1 1 1 0 0 1 0] //lowest bit
3.A←split(A,A<0>) = [4 2 2 5 7 3 1 7]
4.A<1> = [0 1 1 0 1 1 0 1] // middle bit
5.A←split(A,A<1>) = [4 5 1 2 2 7 3 7]
6.A<2> = [1 1 0 0 0 1 0 1] // highest bit
7.A←split(A,A<2>) = [1 2 2 3 4 5 7 7]
```

