### **More About Locality-Sensitive Hashing**

**Question 1**: Consider the following three vectors u, v, w in a 6-dimensional space:  
u = [1, 0.25, 0, 0, 0.5, 0]   
v = [0.75, 0, 0, 0.2, 0.4, 0]   
w = [0, 0.1, 0.75, 0, 0, 1]

Suppose we construct 3-bit sketches of the vectors by the random hyperplane method, using the randomly generated normal vectors r1, r2, and r3, in that order:  
r1 = [1, -1, 1, -1, 1, -1]   
r2 = [-1, -1, 1, 1, -1, 1]   
r3 = [1, 1, 1, 1, 1, 1]

Construct the sketches of the three vectors u, v, w. Estimate the pairwise cosine similarities of u, v, and w from their 3-bit sketches.

**Question 2**: Suppose we have an LSH family *h* of (*d*1,*d*2,.6,.4) hash functions. We can use three functions from *h* and the AND-construction to form a (*d*1,*d*2,*w*,*x*) family, and we can use two functions from *h* and the OR-construction to form a (*d*1,*d*2,*y*,*z*) family. Calculate *w*, *x*, *y*, and *z*.

**Question 3**: This question and the next are based on the following eight strings that represent sets:

s1 = abcef  
s2 = acdeg  
s3 = bcdefg  
s4 = adfg  
s5 = bcdfgh  
s6 = bceg  
s7 = cdfg  
s8 = abcd

Suppose our upper limit on Jaccard distance is 0.2, and we use the indexing scheme of Section 3.9.4 based on symbols appearing in the prefix (no position or length information). For each of s1, s3, and s6, determine how many *other* strings that string will be compared with, if it is used as the probe string. Then, identify the count of s1, s3, and s6.

**Question 4**: Suppose that we have the same 8 strings from Question 3, our upper limit on Jaccard distance is again 0.2, but now we use the indexing scheme of Section 3.9.5 based on symbols appearing in the prefix and the positions of those symbols (no length information). For each of s2, s5, and s7, determine how many *other* strings that string will be compared with, if it is used as the probe string. Then, identify the count of s2, s5, and s7.