# CMPS 101, Spring 2016: HW 4

Merrick Swaffar and Siobhan O'Shea

6 May 2016

**Q1:** The time complexity for matmult(A,x) is $\theta(n^2)$ where A is an nxn matrix and x is an n dimensional vector, because it uses two nested for loops that each do n iterations.

**Q3:** HADMATMULT(H,x)

```
n = x.length
if n<=1, return x
x1 = x[0:n/2]          // split x into x1 and x2
x2 = x[n/2+1:n]
h = H[0:n/2][0:n/2]          // form H_{n/2}
p1 = HADMATMULT(h,x1)
p2 = HADMATMULT(h,X2)
b1 = p1 + p2
b2 = p1 - p2
Return concatenate(b1,b2)
```

The time complexity of hadmatmult(H,x) is $\theta(nlogn)$ where H is an nxn Hadamard matrix and x is an n dimensional vector, because it makes two recursive calls of size n/2 and performs linear time vector additions at each level.

**Q5:** The plot clearly shows that the recursive algorithm is much faster than the brute force method for larger inputs. hadmatmult grows at a rate of about $nlogn$, and matmult grows at a rate of about $n^2$.