:vsp 1a)

i)

Big Theta: f(n) = theta(g(n)) means that there exist a c\_1 and c\_2 such that there exist a N for all x bigger than N we have c\_1\*g(x) <= f(n) <= c\_2 \* g(x)

Big O: f(n) = O(g(n)) means that there exists a *c* such that there exists a *N* for all x bigger than N we have f(n) <= c \* g(x)

Big Omega: f(n) = Omega(g(n)) means that there exists a *c* such that there exists a *N* for all x bigger than N we have c \* g(x) <= f(n)

ii)

T(n) = theta(n)

T(n) = O(n^2)

iii)

false, exponential increases faster than polynomial

False, suppose (3/2)^n <= 2^n^(½), then taking logs of both sides, nlog(3/2) <= n/2 i.e: n\*0.585 <= n/2, this is a contradiction so (3/2)^n >= 2^n^(½) for all n.

true

b)

Don’t think we learnt this.

c)

Don’t think we have covered this.

d) N/A

4a)

vertex: A C D F H E B G I

distance: 0 1 12 20 25 28 34 40 53

b)

c)

i)

Add a source called S which connects all the knights, the edge between the source S to knight i should have capacity q\_i, and we connect i to county j iff i is in the set s\_j, the edge should have capacity q\_i, we then connect all county j to a super sink T, the capacity of the edges should be 1. We then run the max flow algorithm for S and T.

ii)

The runtime of the algorithm is F(v+2, e + m + n)

Alternatively:

Runtime = where i.e. constructing graph + max flow + extract solution.

For curiosities sake:

where is the highest edge capacity under Ford-Fulkerson as it takes O(e) time to find augmenting path, then maximum number of iterations is the highest capacity of a cut which is bounded by eu,