1.

a. the data members could be described as elements of a heap and a map.

b.

k with the highest value in a map would be stored in max_value.
iterate through m(k,v)

return max_value.

c.

increase size by 1

if value is greater than its parents, we don't need to do anything.

Otherwise traverse up tree and fix binary heap.

d.

Get the last element,

Replace root with the last element

Decrease size of heap by 1

Call heapify.

2.

Unionfind will be O(n)

Outer loop will be O(n)

2nd loop will be O(n-i) iters

Inside the 2nd loop will be alpha(n)/iter .

Combine the loops with the content and you get O(n^2*alpha(n))

3.

If(data[n] = 1 or max)

return data(0,mid)+data(mid+1,n)

4.

a

A 2d array.

b. the midpoint would be a good sentinel value. This is because the map will provide a lookup table for values that give the same answer faster, than at an edge.

c.

d. iterate from top to bottom, right to left so that lookup table functions well.

e.

5.

Insert v into heap based on deg(v) will be O(nlgn)

The while loop will be O(lgN+deg(v)/iter)

=lgn+deg(v1)+lgn+deg(v2)_..._lgn+deg(vn)

=N*lgn+sum(deg(v))

=O(nlgn+m)