Part 1. Write a report

a) **Gale-Shapley hospital optimal**

Initially all hospitals and residents are not matched

while there exists a hospital with empty slots and hasn’t asked every resident

choose such a hospital h

let r be the highest preferred resident for hospital h that h hasn’t already asked

if(r is not matched with another hospital)

then (r,h) become matched

else //this means that r is currently matched with h’

if(r prefers h’ over h) //we can make this comparison in constant time by

h remains unmatched. //creating an inverse preference list in the very

//beginning which would take O(n) time for each

// resident

else // r prefers h over h’

(r,h) become matched

h’ gets unmatched

b) the runtime complexity of this algorithm is O(mn). There are at most m hospitals and n residents. There are m\*n possible matches between hospitals and residents, so there are at most m\*n iterations. To maintain this, I initially create an inverse preference list for each hospital which takes a set up time of O(n), but this would allow our comparisons to find if a resident prefers a new hospital over the one its currently matched with in O(1) time.

c) **Gale-Shapley resident optimal**

Initially all residents and hospitals are not matched

While there exists a resident that isn’t matched and hasn’t asked every hospital

choose such a resident r

let h be the highest preferred hospital for resident r that r hasn’t already asked

if(h is not matched with another resident)

then (r,h) become matched

else //this means that h is currently matched with multiple residents r’

find the least preferred resident r’ that hospital h is currently matched with

we can do this using the inverse preference list of the hospital

if(h prefers r over r’) //we can make this comparison in constant time for

(h,r) become matched //each slots the hospital has by creating the

r’ gets unmatched //inverse preference list in the very beginning

// which would take O(n) time for each hospital

else // h prefers r’ over r

r remains unmatched

d) the runtime complexity of this algorithm is O(mnx), where x is the max number of slots any hospital has. There are at most m\*n possible matches between residents and hospitals, so there are at most m\*n iterations. To maintain this, I initially create an inverse preference list for each resident which takes a set up time of O(m), but this would allow our constant time comparisons to find if a hospital prefers a new resident over the least favored resident the hospital is currently matched with. Finding the least preferred resident a hospital is currently matched with takes linear time O(x), the number of slots the hospital has. I was not able to optimize it further other than make the comparison between the least favored resident and the new one in constant time, by using the inverse preference list.