a) Dropping a sheet of glass from a particular floor (x) can yield two cases: The sheet of glass doesn't crack at all or the sheet of glass cracks somewhat. If the glass sheet cracks after dropping it at a certain floor (X), then a check ensues to see with g-1 glass sheets the result of dropping it from the floor below (X-1). If the glass sheet upon dropping it doesn't shatter somewhat, than we chek floors above (x+1...) So that's g glass sheets and max floor minus current floors to check.

b) (floors=4, sheets=2)

$$gFR(4,2)$$

f, s
floor, sheet
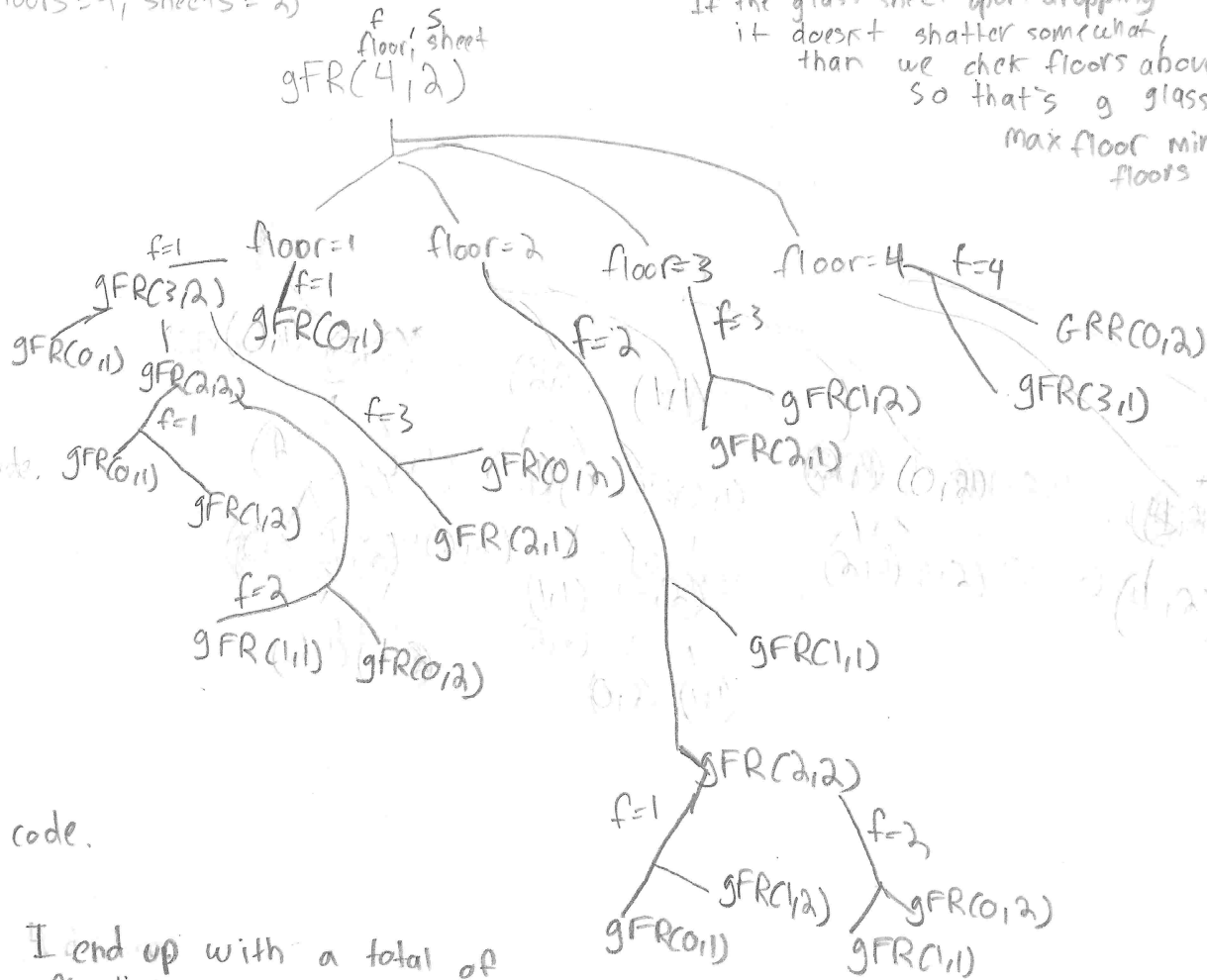
c) code.

d) I end up with a total of 8 distinct subproblems: (3,2),(2,2),(1,2),(0,2),(2,1),(3,1)(4,2) and (1,1).

e) There will be n times m distinct subproblems. As you can see with n(floors)=4 times m(sheets)=2, I remedied 8 distinct subproblems.

f) I would memoize GFR by using an array or a hash table to store the values of the result of using a "top-down with memoization." The recursive procedure would (first) chek to see whether this subproblem has already been solved. If saved and solved, then extract it for use. If not, then save the subproblem to store the value in the standarized data structure.

g) code.