Para empezar empezamos por dibujar un tablero en papel y comenzar a comprender como interactuaba cada llamada a la funcion con respecto a la otra,

nos dimos cuenta que la mejor alternativa era crear una funcion que nos dijiera donde estaba la proxima alternativa con respecto a nosotros, esta nos retorna un numero de -1 al 7 el -1 es usado para cuando no se encuentra una opcion valida a nuestro alrededor, y del 0 al 7 referencia a nuestras casillas vecinas 0 representa la derecha, uno la esquina inferior derecha... y va rotando en el sentido de las aguajs del reloj, luego de saber la siguiente posicion a mirar nos desplazmos hacia ella, y preguntamos si la casilla en la que estamos parados es la letra que buscamos, logicamente esta es la letra que buscamos por la forma que lo implementamos con nuestra funcion que busca la letra en nuestros vecinos. Pero esta comprobacion es para estar seguros de que esta sea la letra que buscamos y no estamos parados en una casillas distinta que puedo haber ocurrido con las rotaciones de la matriz.

al comprobar que esta letra sea la buscada la suplantamos por un ‘-’ para saber que por ese sitio ya fuimos y no volver a tomar en cuenta esa posicion, eso si no sin antes guardar que letra suplantamos en una variable llamada oldChar, esta sera usada mas adelante para deshacer nuestro paso, ademas de suplantar nuestro string, con un nuevo string pero sin la letra que encontramos , antes de salir de nuestro if comprobamos si el largo se nuestra secretWord igual a 0 para corroborar si encontramos toda la palabra secreta que necesitamos. en caso de ser asi se retornara verdadero referenciando que nuetro agente necesita ayuda.

ya sea que esta es la letra que necesitamos o no, se sigue con la busqueda en nuestras casillas vecinas y comprobando si debe hacerse una rotacion. La funcion encargada de las rotaciones recoge el momento en el que se encuantra el algoritmo junto con nuestro mapa y el tama;o de filas y columnas de dicha matriz y dependiento de la situacion plateada en el proyecto este realiza la rotacion necesaria en cada caso.

Con respecto a retorno de cada llamada recursiva, luego se busca en que otra direccion se puede encontrar nuestra letra que no sea la ya visitada, pasando luego a borrar el paso realizado anteriormente suplantando el guion que colocamos anteriormente con nuestra variable oldChart ademas se restarle a nuestro contador. como estamos suplantando el caracter en el que estamos parados no hay problema de que busquemos en una direccion que ya probamos anteriormente, ya que buscamos a partir de nuestras letras vecians.

cada llamada recursiva retorna un bool llamado find dicho find se inicializa en cada llamada en false y se modifica el realizar cada llamada con el return de nuestra funcion la unica forma que retorne false es que la letra no se haya encontrado en ninguna de las direcciones ya que no se habra actualizado nuestra variable find, por ello es importante que despues de ejecutar la llamada recursiva borremos el paso que hemos dado y busquemos la proxima direccion para asi asegurarnos de probar todas las alternativas.

La forma resumida en la que funciona este algoritmo es muy sencilla ya que separamos todo en funciones individuales. Tenemos la funcion readLetter que se encarga de el backtracking aplicando fuerza bruta para conseguir la combinacion deseada, la funcion searchFromSide encargada de decirnos la posicion de la proxima letra con respecto a nosotros, la funcion rotate encargada de aplicar rotateRow o rotateColumn segun sea el caso una funcion que identifica si un numero x pertenece o no a la sucesion de fibonacci, ejecutando la funcion fib mientras que fib(i)<n en el moemnto en el que sea mayor se cancela la ejecucion y se retorna si n== fib en caso de no serlo nuestro n NO PERTENCE a dicha sucesion.

ademas de eso tenemos 2 funciones adicionales que no aportan nada a la resolucion de codigo pero nos parecio importante dejarlas ahi ya que la usamos la mayor parte del tiempo, el print y printFlags, print flags nos imprime nuestra matris pero se;alando la casilla en la que estamos parada y la casilla la cual queremos avanzar asi podiamos ver las interaciiones que haciamos con la matriz. Al igual que una funcion llamada initializeAgente encargada de pedir toda la informacion referente a nuestros agentes. ademas de una funcion de localizar la primera letra de nuestra carta, para evitarnos incluirla en la funcion backtracking ya que podria dificultar el proceso