PRACTICA FINAL ED

Víctor Moreno , Antonio García

La solución a la creación del árbol propuesta nuestra se basa en la llamada a una subfunción que se llamará recursivamente. Vamos a explicar paso a paso.

Vamos a comenzar explicando la función inicial de crear_arbol. Para empezar declaramos un entero que nos servirá para recorrer el vector de atributos y poder saber el atributo a añadir en cada nodo. Hemos optado por utilizar un set de enteros para almacenar la posición de las filas en la que nuestros personajes se encuentran. A continuación creamos nuestro árbol final con la pregunta inicial, la cual la hemos construido a partir del primer atributo y con el tamaño del vector de personajes. El ultimo paso en esta función es crear un subárbol a partir de nuestra función principal crearSubarbol y asignar este subárbol por debajo de nuestro nodo raíz.
Nos vamos a centrar ahora en la explicación de la principal función, crearSubarbol. Esta función va a recibir tres parámetros, el primero será el nodo, el segundo será el numero de pregunta y por ultimo recibirá el set de enteros donde guardaremos la posición de cada jugador, su fila. Lo primero que hacemos cuando accedemos a la función es crear una pregunta con el contenido del siguiente nodo y un árbol auxiliar con esta, después declaramos dos sets de enteros mas y dos arboles mas. Esto es debido a que vamos a diversificar hacia abajo recursivamente. Vamos a ir guardando en un set de enteros los que respondan si a la pregunta y los que respondan no. Los dos arboles serán para la rama derecha y para la rama izquierda. Una vez tenemos todas las declaraciones hechas vamos a diferenciar dos casos en función del numero de personajes restantes. En el caso de que solo quede un personaje restante, hemos llegado a la hoja, en este caso crearemos nuestra pregunta con el nombre del jugador y devolveremos este nodo directamente. En el caso de que queden más de un jugador, significará que no hemos llegado al final del árbol, con lo que debemos seguir bajando. Para ello deberemos declarar dos enteros (personajes_si y personajes_no) en los cuales almacenaremos el numero de personajes restantes para la siguiente pregunta. En este paso, recorremos nuestro vector de filas y vamos almacenando en nuestros nuevos sets las respuestas a cada personaje. Una vez que tenemos en cada set la fila correspondiente a cada personaje y tenemos almacenado el numero de personajes restantes con si y con no, tenemos que recurrir a la recursividad volviendo a llamar a nuestra función, en este caso le pasaremos como argumento el siguiente nodo (derecha para el set de no e izquierda para el set de si) con sus correspondientes sets y numero de personajes. En el caso de que ningún personaje haya respondido si o no a la pregunta formulada, no se creará nodo, con lo que habrá un nodo null en esa posición del árbol. Este bucle se repetirá una y otra vez hasta que alcancemos todos los nodos hoja del árbol.

La otra función principal de esta practica es la función iniciar_juego(), que consiste en un bucle donde el usuario irá respondiendo las preguntas si o no del personaje que el ha pensado y el programa le adivinará el personaje. Básicamente en cada iteración se recorre el árbol y dependiendo si la respuesta es afirmativa o negativa se desplaza al hijo izquierda o al hijo derecha. Si el nodo es un personaje, imprimimos por pantalla el nombre ya que el juego se ha terminado en este punto, sino lo es seguiremos avanzando.