

Informática

Trabajo Final

Ingeniería en Mecatrónica

1. Ordenamiento y búsqueda

Implemente los algoritmos de ordenamiento bubblesort y quicksort, y el algoritmo de búsqueda binaria. Implemente cada uno de ellos en una función. Utilice recursión para implementar quicksort y búsqueda binaria. Escriba además un programa (main) que:

1. genere un arreglo aleatorio de 1000 elementos, cada uno tomando un valor entre 0 y 1000 (utilice la función rand() y el operador módulo %);
2. guarde el valor 500 en una posición aleatoria dentro del arreglo;
3. ordene el arreglo mediante bubblesort o quicksort (a elección del usuario);
4. y finalmente que busque el elemento 500 dentro del arreglo especificado por el usuario y muestre por pantalla el subíndice donde se encontró.

2. Listas enlazadas

Implemente una biblioteca con funciones para manipulación de listas doblemente enlazadas. Cree un nodo para la lista mediante struct (el contenido de cada nodo es a libre elección del alumno), e implemente funciones para

1. agregar un elemento a la lista,
2. eliminar un elemento de la lista,
3. modificar un elemento de la lista
4. mostrar todos los elementos de la lista.

3. Problema de búsqueda de rutas

En el ejercicio 28 de la guía de trabajos prácticos, se solicitó que escribiera una función para leer un archivo con el siguiente formato:

Mapa a ser recorrido (1=ocupado, 0=libre)

```
1000011111
1100010000
0100010000
0100010000
0100010000
0100011000
0100001111
0100000001
0100000001
0111111110
```

#posiciones inicial y final

inicio=(2,3)

fin=(8,8)

Considere que los valores del mapa, su tamaño, las posiciones inicial y final pueden cambiar (es decir, cada combinación de valores constituye una **instancia** del problema).

Ejercicio

Repita el ejercicio utilizando estructuras de datos dinámicas para la matriz. Luego, recorra el mapa leído, realizando un mapping a una estructura de árbol, donde la raíz del árbol es el punto de partida dado por el par (inicioX, inicioY), y donde los "hijos" de cada nodo son las celdas del mapa a las que el agente se puede "mover". Las celdas con valor 0 son espacios libres, mientras que las celdas con valor 1 son obstáculos. La búsqueda (es decir, el recorrido del árbol) debe continuar hasta que se llega a la posición final, dada por el par (finX, finY), o bien hasta que se ha recorrido todo el árbol y no se ha encontrado una solución.

Imprima el camino recorrido por pantalla una vez concluida la búsqueda, o muestre un mensaje de error si no hay tal camino.

Consideraciones

1. Utilice una lista enlazada para llevar el seguimiento de los nodos del árbol recorridos, para imprimirlos al encontrar la solución
2. Utilice recursión para recorrer el árbol