A Simplified Explanation about Data Structures

Resumen del Post de Karuna Sehgal.

Las estructuras de datos son diferentes maneras de almacenar un dato. De esta forma se almacena y ordena de tal manera que nos permite realizar operaciones con los datos de forma efectiva.

La mejor forma de enfrentar un algoritmo es entender como funcionan las diferentes estructuras de datos ya conocidas. además, se agregará una pequeña implementación en Java en cada una.

Array:

Es una estructura de datos que contiene un grupo de elementos. Cada elemento tiene un índice asociado para poder acceder a el. Normalmente todos los elementos son del mismo tipo de dato.

```
1. Int[] arreglo = new Int[3];
2. arreglo[0]=1;
3. arreglo[1]=2;
4. arreglo[2]=3;
```

Hash map:

Es una estructura de datos en duplas, que relaciona un dato con una llave.

```
1. HashMap<Integer,String> hash=new
   HashMap<Integer,String>();
2. hash.put(1, "hola");
3. hash.put(12, "adios");
```

Linked list:

Esta estructura es muy parecida al array, la diferencia es que en esta los datos están enlazados unos con otros por punteros.

```
1.  LinkedList<Integer> listaE = new LinkedList<Integer>();
2.   listaE.add(12);
3.   listaE.removeFirst();
4.   listaE.addLast(30);
```

GRAPH:

Esta estructura de datos es una representación de nodos y flechas conectando estos nodos.

```
1. // se uso la libreria jgrapht
2. Graph<Integer, DefaultEdge> g = new
   DefaultDirectedGraph<>(DefaultEdge.class);
3. g.addVertex(12);
4. g.addVertex(13);
5. g.addVertex(14);
6.
7. g.addEdge(12, 13);
8. g.addEdge(13, 14);
9. g.addEdge(14, 12);
```

TREE:

Es una estructura de datos de nodos enlazados que puede ser usada para representar relaciones jerárquicas en elementos.

```
1. TreeSet<Integer> t = new TreeSet<Integer>();
2. t.add(18);
3. t.add(17);
4. t.add(12);
```

QUEUE:

Es una estructura que ayuda a organizar los elementos en un orden particular. Es una estructura FIFO (First in first out), es decir, el primer elemento insertado es el primero en salir.

```
1. Queue<Integer> q = new LinkedList<>();
2. q.add(12);
3. q.add(13);
4. q.add(11);
5. q.remove();
```

STACK:

Es otra estructura de datos que ayuda organizar los elementos en un orden particular. Esta es una estructura LIFO (Last in First Out), en otras palabras, es una pila y solo se puede acceder al elemento de mas arriba. En el caso que se quiera acceder a un elemento que no sea el tope de la pila de deben sacar los elemento arriba de este.

```
1. Stack<Integer> stack = new Stack<>();
2. stack.push(12);
3. stack.push(13);
4. stack.push(14);
5. stack.peek();
```