**THE MOST IMPORTANT ADTs**

1. Study the main aspects of the ADTs **TREE**, **BINARY TREE**, **GRAPH and WEIGHTED GRAPH** described in the following table.

|  |  |  |  |
| --- | --- | --- | --- |
| **ADT** | **DATA** | **OPERATIONS** | **EXAMPLE** |
| Tree | * Set of **nodes** connected with each other in a hierarchical way
* Each element is connected downwards to N nodes (**children**)
* Each element is connected upwards to 1 node (**parent**)
* Exception: the first node has no father (**root**)
* Some nodes have no children (**leaves**)
 | * Insert element (any position)
* Remove element (any position)
* Insertion and removal have to maintain the tree structure
* Search element
 |  |
| Binary tree | * Set of **nodes** connected with each other in a hierarchical way
* Each element is connected downwards to max 2 nodes (**children**)
* Each element is connected upwards to 1 node (**parent**)
* Exception: the first node has no father (**root**)
* Some nodes have no children (**leaves**)
 | * Insert element (any position)
* Remove element (leaf)
* Insertion and removal have to maintain the binary structure
* Search element
 |  |
| Graph | * Set of **nodes** connected with each other through **edges** (or **arches** or **branches**)
* Each node can be connected to N other nodes (N >=0)
 | * Insert element (any position)
* Remove element (any position)
* Insertion and removal of a node affects also the edges connecting it to the rest of the graph
* Search element
* Search a **path** between two nodes
 |  |
| Weighted graph | * Set of **nodes** connected with each other through **edges** (or **arches** or **branches**)
* Each element can be connected to 1 or to N other nodes
* Each edge has a value (**weight**)
 | * Insert element (any position)
* Remove element (any position)
* Search element
* Calculation of the **path** between two nodes using weights
 |  |

1. Complete the table with the ADTs you already know (Recap).

|  |  |  |  |
| --- | --- | --- | --- |
| **ADT** | **DATA** | **OPERATIONS** | **EXAMPLE** |
| List |  |  |  |
| Queue |  |  |  |
| Stack |  |  |  |