TAD: ChainHashTable

CHT = [(k1,k2, k3… k\_n), (v1, v2, v3…v\_n)]

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Inv: All keys must be different.  **Operations**  **Input**   **Output**   |  |  |  | | --- | --- | --- | | chainHashTable | Hash | Array | | insert | Hash x Key x value | Void | | addLast | Hash x Node x position | Void | | search | Hash x key | Value | | delete | Hash x key | Void | |

|  |
| --- |
| chainHashTable() |
| <Constructor> |
| This method creates a new array |

|  |
| --- |
| insert(key, value) |
| <Modifier> |
| This method allows to add elements to the hashtable if there are no elements in it. |

|  |
| --- |
| addLast(node, position) |
| <Modifier> |
| This method allows to add elements to the hashtable if there are more than one element on the hashtable. |

|  |
| --- |
| search(key) |
| <Analyzer> |
| This method allows to search an element in the hashtable. |

|  |
| --- |
| delete(key) |
| <Modifier> |
| This method allows to delete an element in the hashtable. |

TAD: Heap

Head: [v0, v,1, v2, v\_n-1]

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Invariants:   * v\_n belongs to heap * v\_0 priority is equal of higher to v\_i priority, and 0 is lower or higher to i and i is lower to the size of the heap * v\_(i-1)-2 is the father of v\_i * v\_(i/2+1) is the left son of v\_i * v\_(1/2+2) is the right son of v\_i   **Operations**  **Inputs**   **Outputs**   |  |  |  | | --- | --- | --- | | Heap |  | ArrayList | | maxHeapify | Heap x integer | Void | | heapExtractMax | Heap | Value | | heapIncreaseKey | Heap x Value x key | Void | | heapInsert | Heap x Value x key | Void | | buildMaxHeap | Heap | Void | | heapMaxinum | Heap | Value | | delete | Heap x value | Message | | print | Heap | Message | |  |  |  | |

|  |
| --- |
| Heap() |
| <Constructor > |
| This constructor creates the arraylist in which the elements of the heap will be insert. |

|  |
| --- |
| maxHeapify(integer) |
| <Modifier> |
| This methos is allows to place the highest value at the top |

|  |
| --- |
| heapExtractMax() |
| <Analyzer > |
| This method is used to extract the max value of the priority queue and eliminate it |

|  |
| --- |
| heapIncreaseKey(value, key) |
| <Modifier > |
| This method allows to increase the value of the key for one that has more priority |

|  |
| --- |
| heapInsert(value, key) |
| <Modifier > |
| This method allows to insert a value to heap. |

|  |
| --- |
| buildMaxHeap() |
| <Modifier> |
| This method allows to order the priorityqueue |

|  |
| --- |
| heapMaxinum() |
| <Modifier > |
| This method allows to show the maximum value of the priorityqueue |

|  |
| --- |
| delete(goal) |
| <Modifier > |
| This method allows to delete an element in the heap. |

|  |
| --- |
| print() |
| <Analyzer > |
| This method prints the elements inside the heap. |

**Group members:**

Santiago Jose Belalcazar / A00381777

Daniel Escobar Vacaflor / A00381157

Luis Fernando Pinillos / A00381323