Tecniche di Programmazione 2018

Gruppo 20 – Progetto 2

**ESERCIZIO 1**

La classe NewAVLTreeMap, come richiesto dalle specifiche del progetto, mantiene la stessa interfaccia pubblica di AVLTreeMap e memorizza nei nodi il fattore di bilanciamento invece delle altezze.

L’ultimo comportamento viene ottenuto ridefinendo la classe innestata \_Node(TreeMap.\_Node), includendo in essa l’attributo \_balance\_factor al posto di \_height, attributo originariamente presente in AVLTreeMap.

I metodi hook di AVLTreeMap che implementiamo sono \_rebalance\_insert e \_rebalance\_delete, non ridefinendo invece metodi come \_tall\_child e \_tall\_grandchild in quanto abbiamo assunto che il fattore favorleft sia sempre True. Quindi, in caso di scelta del nipote più alto tra due sottoalberi con uguale altezza, il nodo scelto sarà presente nel sottoalbero sinistro.

|  |  |  |
| --- | --- | --- |
| METODI | IMPLEMENTAZIONE | COMPLESSITA’ |
|  |  |  |
|  |  |  |
|  |  |  |

**ESERCIZIO 2**

descr

|  |  |  |
| --- | --- | --- |
| METODI | IMPLEMENTAZIONE | COMPLESSITA’ |
|  |  |  |
|  |  |  |
|  |  |  |

**ESERCIZIO 3**

Descr

|  |  |  |
| --- | --- | --- |
| METODI | IMPLEMENTAZIONE | COMPLESSITA’ |
|  |  |  |
|  |  |  |
|  |  |  |

**ESERCIZIO 4**

descr

|  |  |  |
| --- | --- | --- |
| METODI | IMPLEMENTAZIONE | COMPLESSITA’ |
|  |  |  |
|  |  |  |
|  |  |  |