**Strutture dati utilizzate**

**TreeArrayList (**Aggiunta del campo **Distanza** alla classe **Node)**

**Descrizione algoritmo**

**Analisi tempo teorico**

* leafDistance: | Eseguo una visita generica a partire dalla foglia, che viene considerata come radice dell’albero
* mediumNode:  **-** numeroFoglie = caso medio, caso peggiore| Costo della visita generica più, per ogni foglia, il costo della funzione **leafDistance**
* calculateSubNode: | Nel caso in cui la funzione sia pari, devo eseguire due visite generiche per ottenere il numero di elementi appartenenti ai due sottoalberi; sommando il costo delle due visite, visito tutti i nodi del grafo. Avendo eliminato l’arco tra i due elementi, devo considerarne uno in meno
* backToFather: **Caso medio** -  **Caso peggiore** | Nel caso peggiore, tutti i nodi appartengono al percorso più lungo

**Analisi tempo sperimentale**

L’analisi sperimentale conferma l’analisi teorica.