**BST (Bumbu Saus Teriyaki)**

* **Insert**
* if insertion point is found
* create new vertex
* if value to be inserted < this key
* go left
* else go right
* **Remove**
* search for v
* if v is a leaf
* delete leaf v
* else if v has 1 child
* bypass v
* else replace v with successor
* **PREDECESORR (ini apa masih belom jelas)**
* if this.left != null return findMax(this.left)
* else
* p = this.parent, T = this
* while(p != null && T == p.left)
* T = p, p = T.parent
* if p is null return -1
* else return p
* **SUCCESSOR (ini juga sama aja)**
* if this.right != null return findMin(this.right)
* else
* p = this.parent, T = this
* while(p != null && T == p.right)
* T = p, p = T.parent
* if p is null return -1
* else return p

**HEAP-HEAP HURA**

* INSERT biar tetep heap heap hura :
* A[A.length] = v
* i = A.length-1
* while (i > 1 && A[parent(i)] < A[i])
* swap(A[i], A[parent(i)])
* HEAP SORT :
* for (i = 0; i < A.length; i++)Max() :
* take out A[1]
* A[1] = A[A.length-1]
* i = 1; A.length—
* while (i < A.length)
* if A[i] < (L = the larger of i's children)
* swap(A[i], L)