

Εργασία 2

Άσκηση 1-2-3:

- **BTCreate(maxelem):** Δημιουργεί ένα δέντρο και το επιστρέφει
- **BTHight(tree):** Επιστρέφει το υψος του δέντρου
- **BTSize(tree):** στέλνει πίσω πόσα nodes έχει το tree
- **BTGetRoot(tree):** κάνει return το root
- **BTGetParent(tree, node):** Σε κάθε node έχω pointer που δείχνει στον parent του Node οπότε απλα επιστρεφω το node->parent, στον πίνακα απλα κάνω την πράξη για να βρω τον parent
- **BTGetChildLeft(tree, node):** Σε κάθε node έχω pointer που δείχνει στο δεξί και αριστερό παιδί και απλά τον επιστρέφει.
- **BTGetChildRigth(tree, node):**
- **BTGetItem(tree, node):** επιστρέφει το item του node που του δίνουμε
- **BTInsertRoot(tree, item):** κανει insert το root εκτός αν έχει ήδη
- **BTInsertLeft(tree, node, item):** κανει Insert στα αριστερά του Node που του δίνουμε (εκτός αν έχει ήδη αριστερό παιδί)
- **BTInsertRight(tree, node, item):** το ίδιο
- **BTRemove(tree, node):** αφερεί το node που του ζητάμε εκτος αν εχει παιδι
- **BTChange(tree, node, item):** αλλάζει το item του node που του δίνουμε
- **BTPreOrder(tree, visit), BTInOrder(tree, visit), BTPostOrder(tree, visit), BTLevelOrder(tree, visit):** κάνουν print
- **BTDestroy(tree):** κάνει free() το δέντρο.

Άσκηση 4:

- **CBTGetLast(tree):** ψαχνει το δέντρο και επιστρεφει τον λαστ node(με τους πίνακες απλά το nodes-1)
- **CBTInsertLast(tree, item):** κανει insert στο τέλος του δέντρου (στον πίνακα απλα το βαζουμε στα nodes και μετα τα ανεβαζουμε +1, προφάνως μόνο αν δεν είναι πάνω απο το maxelem)(στους pointers έχω κανει μια συνάρτηση να βρίσκει τον parent του last node και να βλέπει σε ποιό παιδί του θα μπει
- **CBTRemoveLast(tree):**βρίσκει το τελευταίο node και το κάνει free()

Άσκηση 5:

- **BHCreate(maxelem):** Δημιουργεί ένα δέντρο και το επιστρέφει
- **BHIsEmpty(heap):** στέλνει 1 αν εχει nodes
- **BHGetMaxPriority(heap):** κάνει return το priority του root
- **BHGetMaxItem(heap):** κάνει return το item του root
- **BHRemove(heap):** κάνει remove το root(αλλάζοντας τα στοιχεία του με το τελευταίο node)και μετα κατεβάζει το root οσο χρειαστεί
- **BHInsert(heap, priority, item):** βάζει το node στο τέλος του δέντρου και τον ανεβάζει οσο χρειαστεί
- **BHHeapify(n, priorities, items):** φτιάχνουμε ένα νεο δέντρο με τις CBT συναρτήσεις και κάνουμε αναδιάταξη του δέντρου
- **BHDestroy(heap):** καλεί τον BTDestroy.

Άσκηση 6:

Έκανα την Search οπου βρίσκει το Node που θέλουμε (πηγαίνοντας αριστερά ή δεξιά ανάλογα την τιμή του). Έκανα την Insert οπου βρίσκει που θα μπει το Node (πηγαίνοντας αριστερά ή δεξιά ανάλογα την τιμή του) και μετα βαζοντάς το.

Εκανα την Delete που βρίσκει το node που θέλουμε να κάνουμε delete (καλώντας την search) και μετά ανάλογα την περίπτωση "φτιάχνει" το δέντρο ώστε να μείνει BST!