

Obligatorisk oppgave nr.2

IDATG2102 – Algoritmiske metoder, høsten 2023

Innleveringsfrist: 26.september 2023 kl.11:00 (*må overholdes*)
via Blackboard

Bruk og se filen OBLIG2.cpp/java for hovedrisset til oblig'en.

Det skal lages/kodes to *helt uavhengige* funksjoner. Se i filene om hva en `Node` inneholder, den globale variabelen `gRoot` test-treet som bygges opp/lages, og hvordan de to funksjonene kalles/starter opp. Treet som traverseres er *ikke* sortert på noe kriterie.

NB: Det skal *ikke* innføres flere globale data, datamedlemmer eller parametre enn angitt i de to filene. Det skal heller *ikke* brukes andre hjelpestrukturer - som f.eks. array, stakk, kø, map eller liste.

a) Lag den rekursive funksjonen

```
void skrivIntervall(const Node* node, const int min, const int max)
```

Funksjonen sørger rekursivt for at alle noder som har en `ID` større eller lik `min`, og mindre eller lik `max` får skrevet ut sin verdi på skjermen.

b) Lag den rekursive funksjonen

```
void skrivVenstreBladnode(const Node* node)
```

Funksjonen sørger rekursivt for å skrive ut på skjermen `ID`en til *alle venstre bladnoder* i treet. Prøv å være effektiv, ved å ikke traversere/gå nedom bladnoder som allerede er sjekket.

Kun koden (C++ eller Java) for de to funksjonene leveres som PDF i Blackboard

FrodeH

Obligatorisk Oppgave 2 Løsning

Oppgave a):

```
/**
 * OPPGAVE A: Skriver ut ALLE noder med ID i intervallet 'min'-'max'
 *
 * @param node - Noden som skal besøkes/undersøkes
 * @param min - minste verdi for en utskrevet node sin ID
 * @param max - største verdi for en utskrevet node sin ID
 */
void skrivIntervall(const Node* node, const int min, const int max) {
    if (node) {
        if (node -> ID >= min && node -> ID <= max) {
            printf("\n Node nummer %d ", node->ID);
        }
        skrivIntervall(node -> left, min, max);
        skrivIntervall(node -> right, min, max);
    }
}
```

Oppgave b):

```
/**
 * OPPGAVE B: Skriver alle VENSTRE BLADNODER i treet.
 *
 * @param node - Noden som skal besøkes.
 */
void skrivVenstreBladnode(const Node* node) {
    if (node -> left) {
        if (!node->left->left && !node->left->right) {
            printf("\n Node nummer %d ", node-> left -> ID);
        } else {
            skrivVenstreBladnode(node->left);
        }
    }
    if (node -> right) {
        skrivVenstreBladnode(node->right);
    }
}

// Husk: Det holder ikke å bare sjekke at en node er en bladnode, og så
// skrive dens ID. For det er ikke sikkert at denne noden selv er et
// venstre barn av sin mor
```