

# Trær – Case 1:

( Teksten er hentet fra oppg.nr.3 på eksamen 20/12-2011 i IMT2021-Algoritmiske metoder )

Vi har et binært tre der alle keyer til venstre for enhver node er mindre enn nodens egen verdi, og alle til høyre for enhver node er større eller lik nodens egen verdi (dette kalles egentlig et *binært søketre*, som vi skal lære mye mer om senere). Nodene i treet består av:

```
struct Node {
    int ID; // Nodens ID/key/nøkkel/navn (et tall).
    Node* left; // Peker til venstre subtre, evt. nullptr om tomt.
    Node* right; // Peker til høyre subtre, evt. nullptr om tomt.
    Node (const int id, Node* l, Node* r)
        { ID = id; left = l; right = r; }
};
```

Vi har også den globale variabelen:

```
Node* gRoot = nullptr; // Rot-peker.
```

I *hele* denne oppgaven skal det *ikke* innføres flere globale variable eller struct-medlemmer enn det angitt ovenfor. **Hint:** Tegn opp et litt større tilfeldig binært (søke)tre, så er det lettere å studere/tenke på hvordan de ulike funksjonene skal operere. Treet *kan* inneholde duplikate noder (dvs. flere noder kan ha lik ID).

**a) Lag den ikke-rekursive funksjonen** void hoyreGren()  
Funksjonen skal skrive ut ID'en til *alle* nodene som ligger på stien fra rota og ned til den bladnoden (har to nullptr) som ligger *lengst til høyre av alle bladnoder*. Husk at en bladnode trenger *ikke* å være høyre-barn av sin mor.

**b) Lag den ikke-rekursive funksjonen** void gren(Node\* node)  
Parameteren node peker til en helt *vilkårlig bladnode ett eller annet sted i treet*. Funksjonen skal skrive ut ID'en til *alle* nodene som ligger på stien fra rota og ned til denne bladnoden. Husk at siden treet *kan* inneholde duplikate noder, så *kan* det finnes noder lengre opp i treet som har samme ID som den bladnoden vi leter etter. Funksjonen skal også håndtere dette.

**c) Lag den rekursive funksjonen** void grener(Node\* node)  
Funksjonen skal sørge for at stiene ned til *alle* bladnoder i *hele* treet blir skrevet ut. Lag et linjeskift i utskriften etter at hver sti er skrevet. Kalles fra main ved: grener(gRoot);

**Hint:** Du bør antagelig få bruk for *minst en* av funksjonene laget i oppgave a) eller b)