

Datastructuren Programmeeropdracht 3: Het DSW Algoritme

Deadline: Zondag 3 November, 23:59

In opdracht 2 moest er een “Binary Search Tree” (BST) worden geïmplementeerd. De BST bevat inmiddels de functies `insert`, `search` en `toDot`. In deze opdracht breiden we de functionaliteit uit met:

- `delete`, om een gegeven element uit de BST te verwijderen.
- `getHeight`, bepaalt de hoogte van de BST.
- `getNumNodes`, bepaalt het aantal nodes in de BST.
- `rotateLeft` en `rotateRight`, beide uiteraard `private`.
- `balanceDSW`, balanceert de BST volgens het DSW algoritme. (zie: “Data Structures and Algorithms in C++” van Adam Drozdek)

Invulling van parameters en return waardes is naar eigen inzicht.

Het balanceren moet voor deze opdracht in spiegelbeeld verlopen. Dat wil zeggen dat er eerst een backbone met kinderen naar links gemaakt moet worden. Vervolgens, na het balanceren, ontstaat dan een rechts aangeschoven complete BST.

Demonstreer de werking aan de hand van de volgende operaties:

- Maak een BST aan en vul deze met 31 random integers in $[0, 99]$
- Exporteer de BST naar een DOT bestand
- Balanceer de BST via het DSW algoritme
- Exporteer de BST opnieuw naar een tweede DOT bestand
- Verwijder 10 willekeurige integers uit de BST
- Exporteer de BST naar een derde DOT bestand

Daarnaast dient er een verslag te worden geschreven (in \LaTeX) waarin je de efficiëntie van een ongebalanceerde BST vergelijkt met de efficiëntie van een gebalanceerde BST. Dit gebeurt aan de hand van de Average Internal Path Length (AIPL). Dit is het gemiddelde van de padlengtes naar elk van de knopen van de BST. Kortom, de AIPL is de IPL/n , waarbij n het aantal knopen van de BST is. Deze maat geeft een idee van de complexiteit voor het zoeken binnen een BST, aangezien de padlengte naar een knoop gelijk

is aan het aantal benodigde vergelijkingen wanneer deze knoop opgezocht wordt. Let erop dat de structuur van een ongebalanceerde BST afhangt van de betreffende waardes en de volgorde waarin die zijn ingevoerd. Bereken dus de AIPL voor een divers aantal BST's en neem hier het gemiddelde van.

Om een duidelijk beeld te geven van de gevonden resultaten, moeten deze in een verslag in een aantal grafieken bij elkaar worden geplaatst. Deze grafieken kunnen bijvoorbeeld gegenereerd worden met behulp van GnuPlot. Zie de documentatie over hoe de invoerbestanden aangeleverd moeten worden. Over de x -as kan je het aantal knopen plaatsen en over de y -as de AIPL. Maak in elk geval een grafiek waarbij de optimale gebalanceerde AIPL uitgezet wordt tegen die van de random permutaties.

De uitwerking dient uiterlijk zondag 3 november, om 23:59, via email opgestuurd te worden naar Jan van Rijn (j.n.van.rijn@liacs.leidenuniv.nl).