

Examen voorbeeldvragen

Hoofdstuk 3 Arrayvarianten: set, stack en queue

- Herschrijf de code van stack waarbij je de pointer naar het laatste element laat verwijzen.

CODE VAN CURSUS

```
public class Stack {
    int[] data;
    int pointer=0; // wijst naar het eerste vrije element

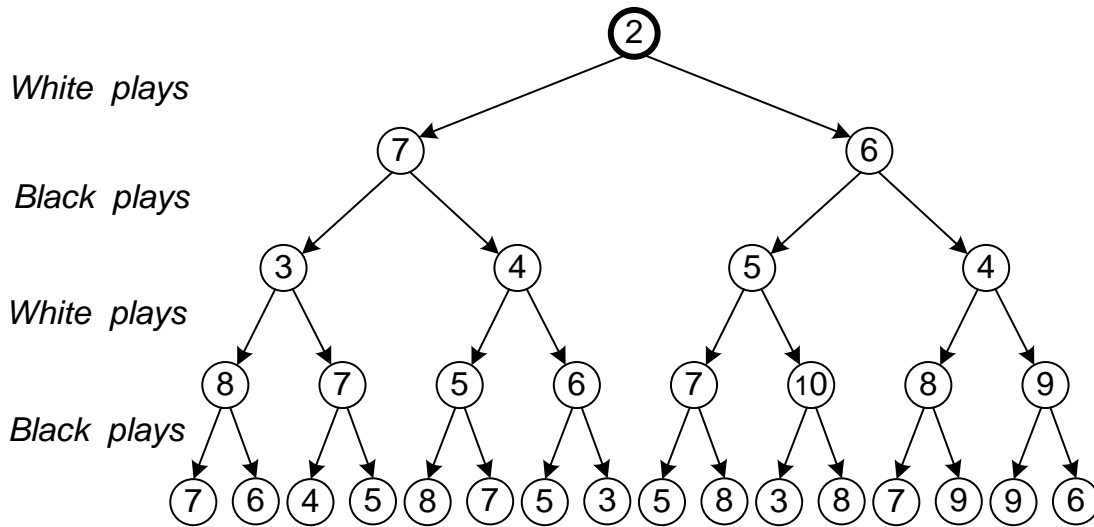
    public Stack(int capacity){
        data = new int[capacity];
    }
    public void push(int element){
        // check of vol
        if (pointer == data.length)
            throw new BufferOverflowException();
        data[pointer] = element;
        pointer++;
    }
    public int pop(){
        // check of leeg
        if (pointer == 0)
            throw new BufferUnderflowException();
        pointer--;
        return data[pointer];
    }
    public boolean isEmpty(){
        return pointer == 0;
    }
    public boolean isFull(){
        return pointer == data.length;
    }
}
```

- Schrijf een methode voor de Stack & FifoQueue die aangeeft hoeveel vrije plaatsen er nog zijn.

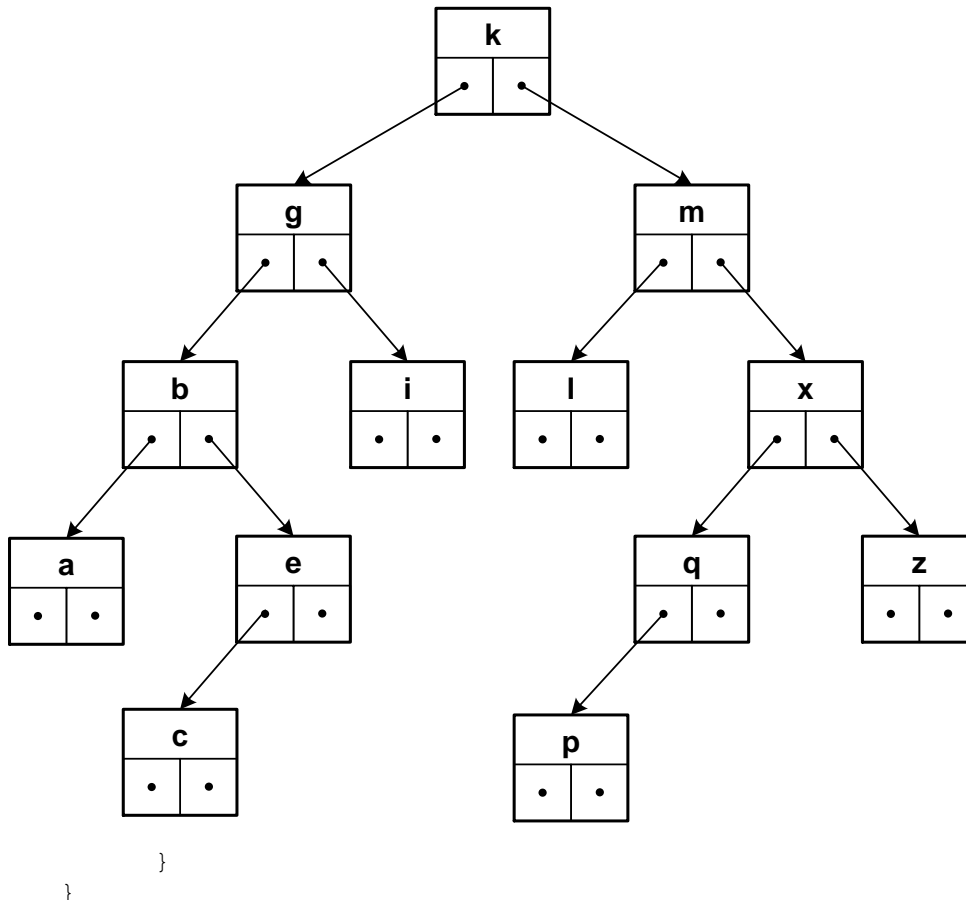
Hoofdstuk 5 Slimme algoritmen

- Gegeven een beurtspel voor 2 spelers waarbij om beurten de spelers een keuze moeten maken tussen L & R. De volgende boom toont de score voor wit voor 4 zetten. Stel dat wit d zetten vooruit denkt en de zet kiest die tot de hoogste score leidt (en zwart voor de kleinste). Welke keuzes zullen wit en zwart maken voor elke waarde van d (1, 2, 3, 4) en waar kom je terecht? (Beide spelers gaan er natuurlijk van uit dat de andere speler ook vooruit denkt.)
 - Wat is je conclusie ivm d ?

- Wat gebeurt er als wit 1 zetten vooruit denkt, maar zwart maximaal vooruit denkt?
- Wat is de rekentijd als je d beurten vooruit wilt denken? Hoe kan je dit optimaliseren (minder rekentijd maar toch ver genoeg vooruit denken)?



- Gegeven een probleem waarbij je een sequentie van zetten moet zoeken die tot de oplossing leidt. Elke zet bestaat uit LEFT, UP, RIGHT of DOWN. Gegeven de volgende boom die alle mogelijke sequenties van 4 zetten weergeeft voor een specifieke instantiatie van het probleem. Waarbij de zwarte node de eindoplossing aanduidt. Er is ook een score gedefinieerd voor elke spelsituatie (de afstand tot de oplossing, die we wensen te minimaliseren). Beschrijf enkele algoritmen die niet gebruik maken van de score alsook een snel algoritme die het wel doet. Bepaal de minimale diepte/horizon om de oplossing te vinden waarbij je er van uit mag gaan dat alle instantiaties van hetzelfde probleem een gelijkaardige boom heeft.



- Gegeven de elementen {c, f, k, q, a, b, z, m, x, p} die we 1-voor-1 toevoegen aan een initieel lege binaire gesorteerde boom. Welke boom is het resultaat?

Hoofdstuk 8 Sorteren

- Gegeven een array waarin alle elementen reeds geordend zijn. Zullen de sorteeralgoritmen deze array sneller sorteren dan een willekeurig-geordende array? Wat is de snelheid in functie van de grootte n voor het algemene geval en voor dit geval? Bespreek dit voor de 5 sorteeralgoritmen van de cursus. Welke is de snelste? Je mag werken met het volgende voorbeeld?
 - {a, b, c, d, e, f, g, h}
- Sorteer de array {3, 6, 8, 1, 5, 2, 4, 7} m.b.v. selection sort, insertion sort, quicksort, recursieve mergesort en natuurlijke mergesort. Geef alle stappen in detail. Is het aantal uitgevoerde vergelijkingen volgens de theoretische verwachting?