Labo opdracht 1

5 oktober 2018

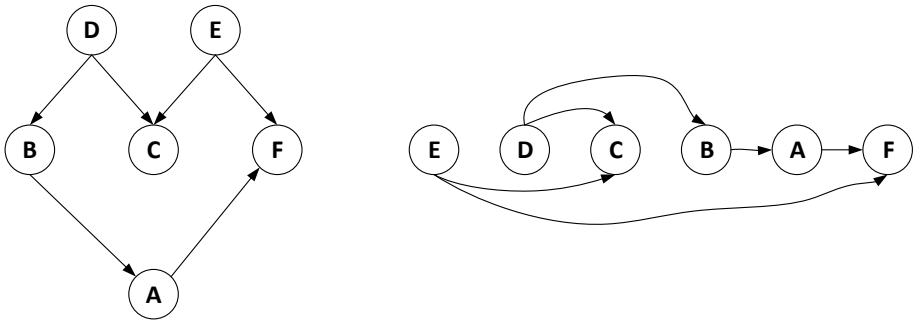
Voor de opgaves kan je eventueel gebruik maken van volgende naslagwerken waar je zeker enkele tips vindt:

- *Introduction to Algorithms, 3rd edition.* Cormen, Leiserson, Rivest, Stein.
- *Algorithm Design.* Jon Kleinberg and Eva Tardos.

We voorzien op Toledo ook al enkele Java klassen, enkele data-bestanden en library *Google Guava* ter ondersteuning van de opgave. Voor opgaves 1) en 2) mag je de gevraagde methodes implementeren in de Java klasse `Labo1DirectedGraphs`. Voor opgave 3) mag je gebruik maken van de klasse `Labo1IntervalTree`, `IntervalTree` en `Interval`.

Gerichte grafen

1. Gegeven is een gerichte graaf $G(V, E)$, met V de set van nodes en E de set van edges. Schrijf een methode die kan bepalen of deze gerichte graaf een (gerichte) lus bevat. We voorzien reeds de klasse `Graph` (onderdeel van *Google Guava*) die alle functionaliteit voor een (gerichte) graaf bevat, samen met een methode die een graaf uit een bestand inleest.
   (2 punten)
2. Een gerichte, acyclische graaf bevat geen lussen. In dat geval bestaat er een sortering van de nodes zodanig dat voor gelijk welke edge (u, v) in de graaf de node u voor de node v in de sortering komt. Zie Figuur 1 voor een voorbeeld.
   (2 punten)

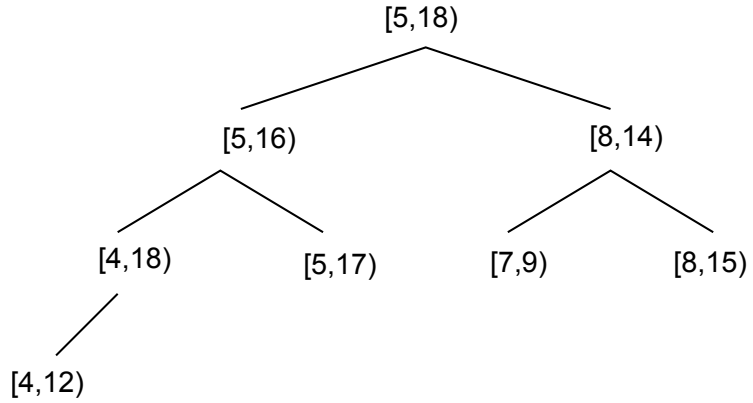


Figuur 1: Ordening van de nodes a, b, c, d, e, f. Merk op dat deze ordening niet uniek is.

Schrijf een methode die een gerichte graaf als argument heeft, en die bepaalt of deze lussen heeft (zie vorige opdracht). Zoja, print een melding van dit feit als resultaat; zonee, print dan de gesorteerde lijst van nodes.

Intervalbomen

Een interval-boom is een datastructuur die intervallen bevat in een gebalanceerde, binaire boom. In deze boom zijn de intervallen van links naar rechts 'gesorteerd', volgens 1) hun ondergrens en 2) indien gelijk, hun bovengrens. Een interval a dat 'kleiner' is dan een interval b bevindt zich dan ofwel in de linker zijtak van de node voor interval b, ofwel bevindt b zich in de rechter zijtak van de node voor interval a. Zie Figuur 2 voor een voorbeeld.



Figuur 2: Voorbeeld: een interval-boom voor de half-open intervallen $[4, 12)$, $[4, 18)$, $[5, 16)$, $[5, 17)$, $[5, 18)$, $[7, 9)$, $[8, 14)$, $[8, 15)$.

Een interval-boom laat toe om enkele queries te doen op deze structuur, waaronder:

1. Gegeven een waarde x, geef een lijst terug met alle intervallen die x bevatten.
2. Gegeven een half-open interval $[a, b)$, geef een lijst met alle intervallen die overlappen met dit interval.

Implementeer de Java klassen `Interval` en `IntervalTree`:

1. Voorzie een constructor die een (ongesorteerde) lijst van half-open intervallen als argument ontvangt en de gebalanceerde boom opstelt. Elke node bevat een `interval`, referenties naar zijn `parent` -node, `left`-node en `right`-node, en ook een waarde `max` die het maximale eindpunt van de intervallen in deze sub-boom bevat. De referenties kunnen `null` zijn naargelang het om de bovenste node gaat, of om onderste nodes.

   Voor de half-open intervallen voor te stellen voorzie je de klasse `Interval` die twee integers bevat, `low` (inclusief) en `high` (exclusief), met getters/setters en die de Comparable-interface correct implementeert. Bovendien moet je ook de overlap met een ander interval kunnen berekenen.
   (1 punt)
2. Voorzie een methode `printIntervals` die de boom doorloopt en de intervallen in de geordende volgorde naar de console schrijft, 1 interval per lijn.
   (1 punt)
3. Voorzie een methode `findOverlapping` die gegeven een integer argument x, een lijst met alle intervallen terug geeft die x bevatten. Maak hier gebruik van de grenswaarden en de structuur van de boom om te vermijden dat je de hele boom doorloopt.
   (2 punten)
4. Voorzie een overloaded methode `findOverlapping` die gegeven een half-open interval $[a, b)$ alle intervallen teruggeeft die overlappen met $[a, b)$. Maak hier gebruik van de grenswaarden

en de structuur van de boom om te vermijden dat je de hele boom overloopt.
(2 punten)

5. Zorg dat alles goed samenwerkt met de klasse `Labo1IntervalTree` die al deze methodes test.