ALGORITMIEK: opgaven werkcollege 1

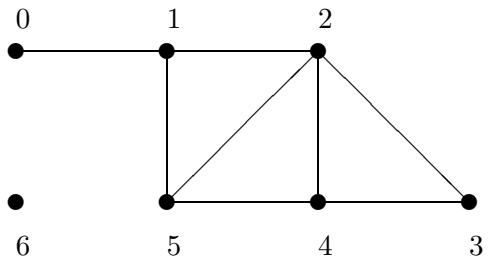
Opgave 1. Kun je een 8 bij 8 schaakbord waarbij één willekeurig wit veld en één willekeurig zwart veld zijn weggelaten volleggen met dominostenen? Zo ja, geef aan hoe. Zo nee, toon dit aan.

Opgave 2. Uit Levitin: opgave 1.1.11. Bewijs ook dat je antwoord (welke lockers zijn na afloop open?) correct is.

Opgave 3. Opgave 2.1.4. (Levitin). Bedoeld ter illustratie van worst/best/average case. Merk op dat linker- en rechterhandschoenen verschillend zijn.

Opgave 4. We hebben n munten, waarvan er één vals is. Dat houdt in dat alle munten behalve de valse evenveel wegen. De valse munt heeft een ander gewicht. Verder zien ze er allemaal precies hetzelfde uit. We hebben bovendien de beschikking over een balans. Probleem: ontdek de valse munt.
a. Een mogelijke oplossing is om de munten twee aan twee te wegen totdat de valse bekend is. Hoeveel wegingen zijn dan minimaal nodig? En hoeveel maximaal?
b. Veronderstel in **b.** en **c.** dat we weten dat de valse munt zwaarder is dan de andere. Geef nu een efficiëntere manier om de valse munt te vinden dan de methode uit **a.** Hoeveel wegingen moet je op die manier minimaal/maximaal doen? Hoeveel is dat voor $n = 24$?
c. Laat zien dat voor $n = 24$ de valse munt in maximaal drie wegingen gevonden kan worden.

Opgave 5. Geef de adjacency-matrix en de adjacency list voor onderstaande ongerichte graaf.



Opgave 6. Leg uit hoe de C++-types voor graafrepresentatie moeten worden geïnterpreteerd dan wel aangepast in geval van gerichte grafen en gewogen grafen.
Geef de adjacency matrix en de adjacency list voor de voorbeeldgrafen **2.** en **3.**

Opgave 7. Geef een algoritme dat bepaalt of er in een gegeven ongerichte graaf hooguit twee knopen zijn met een oneven aantal aangrenzende knopen (buren). Gebruik hierbij de adjacency list om de graaf te representeren.

Opgave 8. Gegeven een gerichte graaf. In de graaf loopt een tak (pijl) van knoop i naar knoop j, maar niet omgekeerd.
a. Schrijf een algoritme dat de tak van richting omkeert. Gebruik de adjacency matrix.
b. Idem, maar nu met de adjacency list.

Opgave 9. Voor de liefhebbers: opgave 1.3.5 en 1.3.8 (Levitin)