ALGORITMIEK: opgaven werkcollege 1

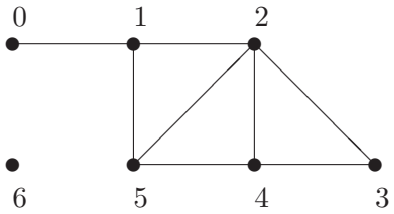
Opgave 1. Kun je een 8 bij 8 schaakbord waarbij één willekeurig wit veld en één willekeurig zwart veld zijn weggelaten volleggen met dominostenen? Zo ja, geef aan hoe. Zo nee, toon dit aan.

Opgave 2. Locker doors (Levitin: exercise 1.1.12)
There are n lockers in a hallway, numbered sequentially from 1 to n. Initially, all the locker doors are closed. You make n passes by the lockers, each time starting with locker #1. On the ith pass, $i = 1, 2, \ldots, n$, you toggle the door of every ith locker: if the door is closed, you open it; if it is open, you close it.
For example, on the first pass, you toggle the doors of all lockers. On the second pass, you toggle the doors of lockers 2, 4, 6, 8, etc. On the third pass, you toggle the doors of lockers 3, 6, 9, 12, etc.
After the last pass, which locker doors are open and which are closed? How many of them are open?
Also proof that your answer (which lockers are open at the end?) is correct.

Opgave 3. (Levitin: exercise 2.1.4)
Meant to illustrate worst/best/average case.
a. Glove selection
There are 22 gloves in a drawer: 5 pairs of red gloves, 4 pairs of yellow, and 2 pairs of green. You select the gloves in the dark and can check them only after a selection has been made. What is the smallest number of gloves you need to select to have at least one matching pair in the best case? In the worst case?
A matching pair is one left glove and one right glove of the same colour. Note that left gloves and right gloves are different.
b. Missing socks.
Imagine that after washing 5 distinct pairs of socks, you discover that two socks are missing. Of course, you would like to have the largest number of complete pairs remaining. Thus, you are left with 4 complete pairs in the best-case scenario and with 3 complete pairs in the worst case. Assuming that the probability of disappearance for each of the 10 socks is the same, find the probability of the best-case scenario; the probability of the worst-case scenario; the number of pairs you should expect in the average case.

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 (steeds één munt op de ene schaal en één munt op de andere schaal) 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 nevenstaande ongerichte graaf.



Bij de hierna volgende opgaven maken we gebruik van de adjacency matrix en de adjacency list representatie van grafen uit het college:

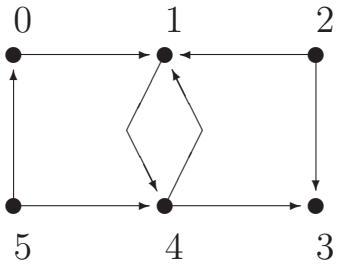
Adjacency matrix

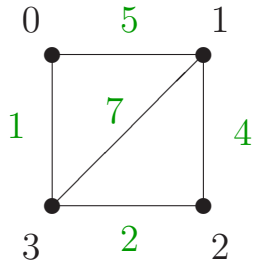
```
int graaf[n][n];
```

Adjacency list

```
class buur
{ public:
    int knoopnummer;
    buur* volgende;
};  // buur
buur* graaf[n];
```

Opgave 6. Leg uit hoe de C++-types voor graafrepresentatie moeten worden geïnterpreteerd danwel aangepast in geval van gerichte grafen en gewogen grafen. Geef de adjacency matrix en de adjacency list voor de voorbeeldgrafen **2.** en **3.** uit het eerste college (hieronder).





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. **a.** Schrijf een algoritme dat het totale aantal takken in een gegeven ongerichte graaf bepaalt. Gebruik de adjacency list representatie.
b. Doe hetzelfde, maar nu met de adjacency matrix representatie.
c. Welke graafrepresentatie is in dit geval het meest geschikt (efficiënt)?

Opgave 9. 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 10. Voor de liefhebbers (over ggd en bekende graafproblemen): opgave 1.1.6, 1.3.4, 1.3.5 en 1.3.8 (Levitin)