1. (a) Om het getal A^N (voor gehele getallen A en N met $N \geq A \geq 1$) te berekenen, kunnen we het volgende, eenvoudige algoritme gebruiken:

```
Macht = 1;
for  (i=1 to N)
{ Macht = Macht * A;
}

write (Macht);
```

   Wat is de tijdscomplexiteit van dit algoritme? Is dit polynomiaal of exponentieel in de grootte van de invoer? Motiveer je antwoorden.

Om het getal A^N (voor gehele getallen A en N met $N \geq A \geq 1$) te berekenen, kunnen we ook het volgende, 'listige' algoritme gebruiken:

```
Macht = 1;
Factor = A;
Exponent = N;
while (Exponent != 0)
{ if (Exponent is een even getal)
  { Factor = Factor * Factor;
    Exponent = Exponent / 2;
  }
  else
  { Macht = Macht * Factor;
    Exponent = Exponent - 1;
  }
}

write (Macht);
```

   (b) Voer het 'listige' algoritme uit voor het geval dat $N = 41$ (het grondtal A houden we algemeen). Dat wil zeggen: voor iedere iteratie van de while-lus, geef aan wat de variabelen Exponent, Factor en Macht zijn aan het eind van de iteratie.

   (c) Laat zien dat de volgende bewering een invariant is voor de while-lus van het 'listige' algoritme. Dat wil zeggen: dat de bewering waar is, iedere keer dat we de test `(Exponent != 0)` uitvoeren:

   $$\text{Factor}^{\text{Exponent}} \times \text{Macht} = A^N.$$

   Doe dit niet speciaal voor het voorbeeld met $N = 41$, maar voor algemene N.

   (d) Gebruik de invariant van het vorige onderdeel om aan te tonen dat het 'listige' algoritme partieel correct is. Dat wil zeggen: om aan te tonen dat *als* het algoritme eindigt, dat dan de uitvoer correct is.

   (e) Geef een passende convergent voor de while-lus van het 'listige' algoritme. Motiveer je keuze voor deze convergent.

   (f) Hoeveel iteraties heeft de while-lus van het 'listige' algoritme in het slechtste geval (als functie van N)? Wat is (dus) de tijdscomplexiteit van het 'listige' algoritme in het slechtste geval? Motiveer je antwoorden.

2. Orden de volgende vijf complexiteiten van mogelijke algoritmes, van goed (snel) naar slecht (langzaam): $\mathcal{O}(2^{(2^N)})$, $\mathcal{O}(N^2)$, $\mathcal{O}(N \times \log(N))$, $\mathcal{O}(5^N)$ en $\mathcal{O}(\log(N))$.

3. Voor het handelsreizigersprobleem kunnen we twee varianten formuleren. Een beslissingsvariant TSPb:

   > Gegeven een ongerichte graaf G met N knopen en M takken, met gewichten op de takken, en gegeven een getal K. Bestaat er een route in de graaf die begint bij een knoop, vervolgens alle andere knopen precies één keer bezoekt en dan in de laatste stap terugkeert bij de beginknoop, met een totaal gewicht van hoogstens K?

   En een optimaliseringsvariant TSPo:

   > Gegeven een ongerichte graaf G met N knopen en M takken, met gewichten op de takken. Wat is het minimale totale gewicht van een route in de graaf die begint bij een knoop, vervolgens alle andere knopen precies één keer bezoekt en dan in de laatste stap terugkeert bij de beginknoop?

   (a) Geef een niet-deterministisch polynomiaal algoritme voor de beslissingsvariant TSPb. Neem hierbij aan dat de knopen nummers 1,2,...,N hebben.

   (b) Hoe kunnen we een oplossing voor de optimaliseringsvariant TSPo gebruiken om de beslissingsvariant TSPb op te lossen? Dat wil zeggen: stel dat we een functie $TSPo(G)$ hebben die de waarde van TSPo oplevert (of oneindig groot als er helemaal geen route is), geef dan pseudocode voor een functie $TSPb(G, K)$ die gebruik maakt van $TSPo(G)$.
   Wellicht ten overvloede: de functie $TSPb(G, K)$ geeft als antwoord `"ja"` of `"nee"`, afhankelijk van de graaf G en het getal K.

   Omgekeerd kunnen we een oplossing $TSPb(G, K)$ voor de beslissingsvariant, gebruiken om de optimaliseringsvariant op te lossen, met het volgende algoritme:

```
if (TSPb(G,MaxGewicht))
{ while (MinGewicht != MaxGewicht)
  { Midden = (MinGewicht + MaxGewicht)/2;
    if TSPb(G,Midden)
      MaxGewicht = Midden;
    else
      MinGewicht = Midden+1;
  }
  write (MaxGewicht);
}
else
  write "geen tour mogelijk"
```

   Hierbij nemen we aan dat de gewichten van de takken gehele getallen zijn. En we nemen aan dat de variabelen MinGewicht en MaxGewicht aan het begin bepaalde, goedgekozen, gehele waardes hebben, waarbij MinGewicht $\leq$ MaxGewicht.

   (c) Wat is het aantal iteraties van de while-lus als aan het begin van het algoritme MinGewicht=10 en MaxGewicht=73? Motiveer je antwoord.

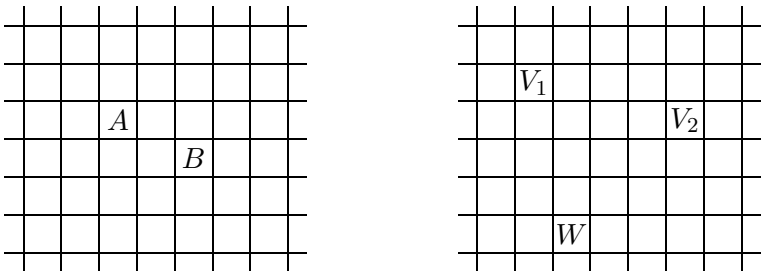
   (d) Wat is het aantal iteraties van de while-lus in het algemeen in het slechtste geval, als functie van de variabelen MinGewicht en MaxGewicht? Motiveer je antwoord.

   (e) Zoals gezegd: de graaf G heeft N knopen en M takken.
   Stel dat het minimale gewicht van een tak in de graaf 1 is en dat het maximale gewicht van een tak in de graaf 10 is. Wat zijn dan optimale waardes voor de variabelen MinGewicht en MaxGewicht? Dat wil zeggen: waardes waarvoor het beschreven algoritme altijd het goede antwoord geeft, maar waarvoor het zo min mogelijk stappen kost? Motiveer je antwoord.

4. (a) Stel dat we de volgende drie tegeltypen hebben:

waarbij 1, 2, 3 en 4 verschillende kleuren aanduiden. Van elk type hebben we oneindig veel tegels. Bestaat er een tegelpad ('domino snake') van vakje A naar vakje B in het linker rooster hieronder?

N.B.: het rooster loopt in alle richtingen oneindig ver door, en het pad mag zo nodig heel grote omwegen maken om van A naar B te komen.



(b) Beschouw het volgende beslissingsprobleem:

> Gegeven een aantal tegeltypen (waarvan we er steeds weer oneindig veel hebben) en drie verschillende vakjes V_1, V_2 en W in het hele platte vlak (oneindig groot in alle kanten) (zie bijvoorbeeld het rechter rooster hierboven). Bestaat er zowel een tegelpad van W naar V_1, als een tegelpad van W naar V_2?

Is dit beslissingsprobleem oplosbaar? Motiveer je antwoord.

Wellicht ten overvloede: het gaat om het beslissingsprobleem voor het algemene geval. Het gaat dus niet speciaal om de tegeltypen van onderdeel (a), en de vakjes V_1, V_2 en W in het rechter rooster hierboven.

(c) We gaan het probleem van het vorige onderdeel aanpassen.

> Gegeven een aantal tegeltypen (waarvan we er steeds weer oneindig veel hebben) en drie vakjes V_1, V_2 en W in een rechthoek in het platte vlak. Bestaat er zowel een tegelpad van W naar V_1, als een tegelpad van W naar V_2, zó dat beide tegelpaden binnen de rechthoek blijven?

Is dit beslissingsprobleem oplosbaar? Motiveer je antwoord.

5. (a) Beschrijf hoe de RSA encryptie-methode werkt. Dat wil zeggen: hoe publieke en geheime sleutels worden afgeleid (zeg ook expliciet wat de publieke sleutels zijn en wat de geheime), en hoe codering en decodering van een boodschap M in zijn werk gaan. Ga er voor het gemak vanuit dat M een geheel getal is met $M \geq 0$.

(b) Pas de methode toe voor de priemgetallen $P = 5$ en $Q = 11$, codeer de boodschap $M = 10$ en decodeer het resultaat weer. Laat duidelijk zien hoe je te werk gaat.

Bij dit onderdeel mag je je rekenmachine gebruiken.