1. Deze opgave gaat over een algoritme om voor een string S de omgekeerde string reverse (S) te construeren. Dat wil zeggen: S van achteren naar voren gelezen. Hierbij maken we gebruik van functies
   – head (X): levert de eerste letter van de string X op
   – tail (X): levert de string X behalve zijn eerste letter op
   – X · Y: levert de concatenatie van de strings X en Y op ('X gevolgd door Y')

   Het algoritme ziet er als volgt uit:

```
X = S;
Y = Λ;   // de lege string dus
while (X ≠ Λ)
do   Y = head (X) · Y;
     X = tail (X);
od

return Y;
```

   (a) Toon aan dat de volgende bewering een invariant is voor de while-lus in het bovenstaande algoritme:

   S = reverse (Y) · X

   Hint: gebruik het feit dat in het algemeen voor twee strings V en W geldt dat

   reverse (V · W) = reverse (W) · reverse (V)

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

   (c) Geef een passende convergent voor de while lus in het algoritme, dat wil zeggen: een uitdrukking die iedere iteratie van de while-lus kleiner wordt, en nooit negatief zal worden.

2. Deze opgave gaat over een simpele variant van het spel Risk. We hebben twee partijen, partij A met $N \geq 1$ legers en partij B met $M \geq 1$ legers. Partij A valt partij B aan.

   Hierbij strijdt steeds één leger van partij A tegen één leger van partij B. Een van de twee legers wint, en het leger dat verliest verdwijnt uit het spel. Dit gaat zo door totdat partij A of partij B geen legers meer over heeft. De andere partij heeft dan gewonnen.

   Wanneer een leger van partij A tegen een leger van partij B strijdt, bepaalt een dobbelsteen wie er wint. Omdat aanvallen moeilijker is dan verdedigen, wint het leger van partij A alleen als er 5 of 6 gegooid wordt met de dobbelsteen.

   We zijn nu benieuwd naar de kans dat partij A met zijn N legers wint van partij B met zijn M legers. Laten we die kans noteren als Winstkans (N, M).

   (a) Beargumenteer dat voor $N \geq 1$ en $M \geq 1$ geldt, dat

   $$\text{Winstkans (N, M)} = \tfrac{1}{3} \times \text{Winstkans (N, M-1)} + \tfrac{2}{3} \times \text{Winstkans (N-1, M)}$$

   Hierbij definiëren we voor de eindsituaties:

   Winstkans (N, 0) = 1, als $N \geq 1$
   Winstkans (0, M) = 0, als $M \geq 1$.

   Leg ook uit waarom we deze twee kansen zo definiëren voor de eindsituaties.

   (b) Bereken (zonder rekenmachine) Winstkans (1, 1) en Winstkans (2, 1). Laat zien hoe je aan je antwoorden komt.

   (c) We kunnen nu voor algemene N en M de volgende recursieve functie gebruiken om Winstkans (N, M) te berekenen:

```
Winstkans (N, M)
  // aanname:  N ≥ 0 en M ≥ 0
  // en minstens een van beide is positief
{    if (M == 0)
     then return 1;
     else if (N == 0)
          then return 0;
          else  return 1/3 × Winstkans (N, M-1) + 2/3 × Winstkans (N-1, M);
          fi
     fi
}
```

   Geef de boom met alle recursieve aanroepen als we de functie Winstkans aanroepen voor N = 3 en M = 2.
   N.B.: je hoeft dus niet Winstkans (3, 2) uit te rekenen, maar je moet alleen laten zien voor welke waardes van N en M de functie steeds (recursief) wordt aangeroepen.

   (d) Om dubbel werk te voorkomen, kunnen we Winstkans (N, M) ook met dynamisch programmeren berekenen. Geef daarvoor een algoritme.

   (e) Wat is de tijdscomplexiteit van je algoritme uit het vorige onderdeel. Motiveer je antwoord.

3. In deze opgave gaan we knopen tellen in binaire bomen. Hierbij zeggen we dat een binaire boom van hoogte 1 gelijk is aan een enkele knoop. Vervolgens is een binaire boom van hoogte 2 gelijk aan een wortel met een of twee kinderen. Enzovoort.

   (a) Geef voorbeelden van binaire bomen van hoogtes 1, 2, 3 en 4 met zo weinig mogelijk knopen erin.

   (b) Geef voorbeelden van binaire bomen van hoogtes 1, 2, 3 en 4 met zo veel mogelijk knopen erin.

   (c) Laat N_h het maximum aantal knopen in een binaire boom van hoogte h zijn (zoals bij het vorige onderdeel dus). Wat zijn achtereenvolgens N_1, N_2, N_3, N_4?

   (d) Beargumenteer dat voor $h \geq 2$ geldt dat $N_h = 1 + 2 \times N_{h-1}$.

   (e) Toon met behulp van inductie aan dat voor elke $h \geq 1$ geldt dat $N_h = 2^h - 1$.

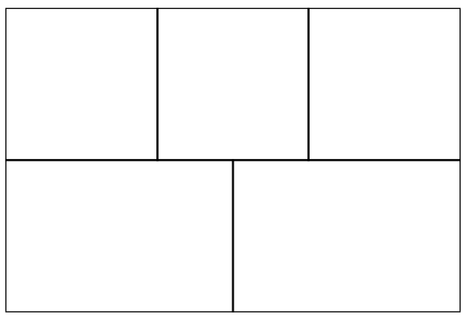
   (f) Een complete binaire boom van hoogte h is een binaire boom van hoogte h, waarbij de eerste $h-1$ niveaus helemaal gevuld zijn met knopen, en de knopen op het h-de niveau 'zo veel mogelijk naar links' zitten.

   Hoeveel knopen bevat een complete binaire boom van hoogte h minimaal en hoeveel knopen maximaal? Motiveer je antwoorden.

4. Landkaarten moeten op zo'n manier gekleurd worden dat landen die aan elkaar grenzen verschillende kleuren krijgen. Landen die elkaar alleen in een punt raken, mogen wel dezelfde kleur krijgen.

   Wanneer er gevraagd wordt om een voorbeeld van een landkaart (bij onderdelen (c), (d) en (e)), moet het een 'samenhangende' landkaart zijn: je moet van ieder land naar ieder ander land kunnen reizen. Er zijn dus geen losse 'eilandjes'.

   (a) Beschouw de volgende 'landkaart':

   Kleur de landkaart met zo min mogelijk kleuren. Leg ook uit waarom de landkaart niet met minder kleuren gekleurd kan worden.

   (b) Geef een voorbeeld van een landkaart met minstens zes landen, die met twee kleuren gekleurd kan worden (en niet met minder). Kleur die landkaart ook in met twee kleuren.

   (c) Geef een voorbeeld van een landkaart met minstens zes landen, die met drie kleuren gekleurd kan worden (en niet met minder). Kleur die landkaart ook in met drie kleuren.

   (d) Geef een voorbeeld van een landkaart met minstens zes landen, die met vier kleuren gekleurd kan worden (en niet met minder). Kleur die landkaart ook in met vier kleuren.

(e) Beschouw de volgende drie beslissingsproblemen:

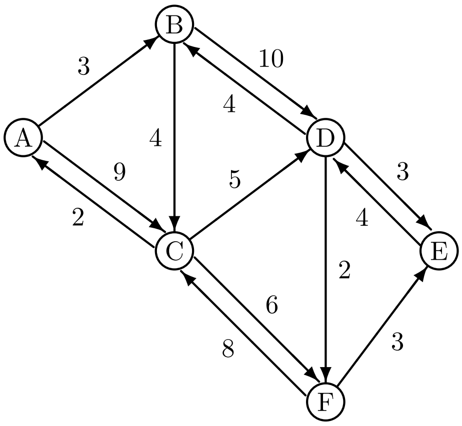
Landkaart-2-kleuring: gegeven een landkaart. Is het mogelijk om de landkaart met 2 kleuren te kleuren ?

Landkaart-3-kleuring: gegeven een landkaart. Is het mogelijk om de landkaart met 3 kleuren te kleuren ?

Landkaart-4-kleuring: gegeven een landkaart. Is het mogelijk om de landkaart met 4 kleuren te kleuren ?

Orden deze drie beslissingsproblemen van gemakkelijk naar moeilijk. Motiveer de volgorde die je kiest.

5. Beschouw onderstaande gerichte graaf met gewichten op de takken:



(a) Stel dat we het kortste pad van knoop A naar knoop E willen berekenen. Waarom kunnen we daarvoor niet het behandelde algoritme met dynamisch programmeren gebruiken?

(b) Bereken het kortste pad van knoop A naar knoop E met behulp van het algoritme van Dijkstra. Leg duidelijk uit hoe je te werk gaat en geef ook tussenresultaten.