Derde serie sommen.

1. (5 punten) Geef een dynamic programming algoritme voor het probleem Langste Stijgende Subrij: Gegeven een rij getallen, vind de langste deelrij waarvan de elementen monotoon stijgend zijn (ze hoeven niet naast elkaar te staan). Dus bijvoorbeeld van $S = (9, 5, 2, 8, 7, 3, 1, 6, 4)$ heeft de langste stijgende deelrij lengte drie en is $(2, 3, 4)$ of $(2, 3, 6)$.

2. (5 punten) Geef een dynamic programming algoritme voor het probleem Maximale Som: Vind in een rij getallen de deelrij van naast elkaar liggende getallen met maximale som (bijvoorbeeld in $\{31, -41, 59, 26, -53, 58, 97, -23, 84\}$ is dat de rij van 59 t.e.m. 84).

3. (10 punten) Een bepaalde chemische stof kan maximaal n keer door een veredelingsproces gehaald worden, en elke keer dat dat gebeurt gaat er wat van die stof verloren. De stof kan ook na een willekeurige stap in dat proces verkocht worden. Als een hoeveelheid x van de stof na stap r verkocht wordt levert dat $\phi_r(x)$ op. Als een hoeveelheid y van de stof door het veredelingsproces gehaald wordt, heb je daarna nog maar $a \cdot y$ van de stof over (hier is $a < 1$ de factor). Bedenk een dynamic programming algoritme die uitrekent hoeveel van de stof je in welke stap moet verkopen om de maximale winst te scoren. Wat is de complexiteit van uw algoritme?

4. (5 punten) Op een conferentie worden werkgroepen gevoerd om een probleem te tackelen. Deelnemers aan de conferentie komen van een aantal verschillende onderzoeksinstituten. Om dominantie van een bepaalde school te voorkomen, wil men een indeling maken zodat geen twee deelnemers van hetzelfde instituut in dezelfde werkgroep terecht komen. Stel dat er p instituten en men q werkgroepen wil vormen. Hoe lost men dat met gebruikmaking van Network Flow op.

5. (programmeren 10 punten). Vergelijk de algoritme van Ford en Fulkerson met de MKM algoritme op een random graph.
   Maak voor een redelijk grote n (begin klein totdat het programma correct is en maak dan n steeds groter om een verschil in complexiteit te krijgen) een random netwerk. Neem integer capaciteiten, zodat tenminste convergentie gegarandeerd is. Het netwerk kan worden gerepresenteerd door een matrix c met $c[i, j] > 0$ als er een kant van i naar j is. Een matrix f met eerst $f[i, j] = 0$ kan de stroom representeren. Door vergelijking van f en c kan voor FFK een stroomvergrotend pad met DFS gevonden worden, totdat zo'n pad niet meer bestaat, voor MKM is minstens *nog* een matrix nodig om het gelaagde netwerk te representeren. Meet met de wallclock of voor redelijk grote n—bijvoorbeeld in de buurt van de 100—al een verschil gezien kan worden in de complexiteit van deze beide algoritmen.