

# Programmerings-NM 2003-09-27

## Kortfattede løsningskommentarer

### A. Hardware

En vanlig registreringsoppgift på grundkursnivå som løses på valfritt sätt - Brute Force.

### B. Budget

Kan reduseras till en maxflow-algoritme.

Bipartit graf, ett hörn  $u_i$  till vänster per rad, ett hörn  $v_j$  till höger per kolumn. Dessutom källa  $s$  och utlopp  $t$ .

Källa  $s$  har kant till varje radhörn med kapacitet = radsumma för den raden.

Utlopp  $t$  har kant från varje kolumnhörn med kapacitet = kolumnsumman för den kolumnen.

För varje ruta  $(i,j)$  i budget-matrisen så finns kanterna  $(u_i,v_j)$  och  $(v_j,u_i)$ .

Om villkoret är att talet i rutan ska vara  $a \leq \text{talet} \leq b$  så sätter vi kapacitet  $b$  på kanten  $(u_i,v_j)$  och kapacitet  $-a$  på kanten  $(v_j,u_i)$ .

Finn sedan maxflöde (med t.ex. Ford-Fulkerson) och kolla att kanterna från  $s$  och kanterna till  $t$  är mättade. Då har vi en tillåten lösning på budgetproblemet.

En komplikasjon är att man måste hitta ett tillåtet startflöde. Det är inte säkert att noll-flödet är tillåtet eftersom vi har villkor som säger att flöde $[v_i,u_j] \geq a$  för vissa  $(i,j)$ .

### C. Frogger

Løses med bredde først søk. For hver runde må man flytte på alle bilene slik at man ser hvor frosken (grodan) kan hoppe. Det letteste (?) er å holde alle posisjonene frosken kan nå i en runde i en 2D tabell av samme størrelse som den "fysiske" verden. Da vet man at man i neste runde kan nå alle de samme rutene + alle naboruter - alle ruter hvor det er en bil. Så snart man har merket av målruten er problemet løst.

### D. Gallup

Løses med "brute force". Prøv først med 1 person, så 2 osv. Hver gang: Regn ut hvor mange personer som har valgt hvert alternativ og regn tilbake prosentatsen og sjekk om det stemmer. Sjekk også summen av antall personer.

Komplikasjon: Pga avrunding kan det være flere antall personer som passer til prosentatsen. Finn største og minste antall og summer disse for å sjekke antallet personer.

Komplikasjon: sprint i C avrunder ikke riktig (nevnt i oppgaveteksten).

## E. Jackpot

Perioden mellan förekomster av "Jackpot" är det minsta heltal som är delbart med periodiciteten för varje enskilt hjul, dvs. hjulperiodiciteternas minsta gemensamma multipel (least common multiple, LCM på Engelska).

En enkel implementation av LCM kan utnyttja att  $\text{LCM}(x_0, x_1, x_2, \dots) = \text{LCM}(x_0, \text{LCM}(x_1, \text{LCM}(x_2, \dots)))$  och att  $\text{LCM}(x, y) = x * y / \text{GCD}(x, y)$ .  $\text{GCD}(x, y)$  beräknas som bekant m.h.a. Euklides algoritmen.

En alternativ, men lite mer krävande lösningsmetod är att primtalsfaktorisera varje hjulperiod, och bygga upp LCM som en faktor av maximala antalet förekomster av varje primtalsfaktor.

## F. Spiderman

Standard dyn-prog. Bra att beräkna  $\text{minheight}[i, h] =$  minsta höjd på bygnad som medger att vi befinner oss på höjd  $h$  efter de  $i$  första etapperna.

## G. Subway

Börja med att plocka bort de punkter som ligger  $\leq d$  från origo. För de återstående punkterna, beräkna vinkel ( $\text{atan2}$ ) från origo till punkten samt inom vilket vinkelintervall (trigonometri) en linje från origo kan ha för att punkten ska vara  $\leq d$  från linje. Sortera alla intervall så att en cirkulär lista uppstår. Bryt upp listan på alla möjliga ställen så att den blir linjär, applicera därefter en greedy-lösning för varje linjär lista.

Vid sortering kan det krävas att man är noga med flyttalsjämförelser. Det går förvisso att lösa endast med heltal, men om man använder ett litet epsilon vid alla flyttalsjämförelser går det bra. Möjliga värden på epsilon som fungerar på min lösning är  $1e-5$  till  $1e-18$  (eller nåt sånt).

## H. CPU

Denna uppgift blir ganska lätt om man bara angriper den från rätt håll. Ett Zeisel-nummer (vilket det egentligen rör sig om) är en produkt av minst tre primtal, och kan i uppgiften aldrig vara större än  $2 * 10^9$ . Detta sätter klara begränsningar på de två lägsta av primtalen. Det räcker således att loopa över alla tillräckligt små unika par av primtal och se om de utgör början av en Zeisel-sekvens. Om de gör det, så lägger man till alla Zeisel-nummer som har det primtalsparet som lägsta faktorer i en lista. När detta är klart har man en lista över samtliga 243 "explosiva" nummer som finns i intervallet. Efter detta är det bara att söka i den listan för att se hur många som finns inom det givna intervallet.