

IMPA Omgång 8 – 2010-11-03

10013 – *Super long sums*

Hur gör man för hand?

Läs in talen till två långa vektorer. Addera de två vektorerna element för element från minst signifikanta siffran. Om summan blir 10 eller mer, lägg till ett på nästa index. Se till att klara av fallet att de mest signifikanta siffrorna summerar till 10 eller mer.

10365 – *Blocks*

Fullständig sökning.

Iterera över alla kombinationer och kontrollera vilka kantlängder som ger minst area. Notera att man alltid kan ordna kanternas längder efter storleksordning. Vidare så kan den tredje sidan beräknas från de två föregående och den givna volymen.

Vilka iterationsgränser behövs? Om alla kanter är lika långa så har de längden 10. Om en sida har minsta längd 1 så kan nästa kant ha längd till 31 ty $32^2 > 1000$. Alltså behövs bara två loopar med 10 respektive 31 som iterationsgräns.

10151 – *Spaghetti*

Det må finnas flera lösningar till detta problem. Dock måste man alltid lyckas att tolka ut etiketter samt if- och goto-satser.

En lösning är att köra memoisering. Kontrollera om programmen är identiska om man står på rad x respektive y . Om det är en normal sats så är programmen lika om anropet `isEqual(x+1, y+1)` också returnerar sant. Vid en if-sats ska båda grenarna vara lika.

Om man i anropet står på en goto-rad så är det bara att räkna framåt tills man inte står på någon sådan rad eller tills att en loop upptäcks. I det senare fallet är programmen ändå lika om båda har loopar.

Det svåra i uppgiften är att inse att om man får ett anrop till `isEqual(x, y)` med samma rader x och y som tidigare i stacken, så ska man anse att programmen är lika. Orsaken är att om det är lika ett varv i loopen så kommer det fortsätta att vara lika hur många gånger loopen än upprepar sig.

Till sist bör man notera att rekursionsdjupet vid sådana här lösningar kan bli väldigt stort och kanske fyller man stacken helt. En lösning kan då vara att implementera en egen stack och inte anropa rekursivt.