Det här är ett udrag ur boken "Matematisk utflykt" skriven av Erik Svensson och Valentina Chapovalova. "Logiker med hattar" är ett av problemen på nivå 3.

Logiker med hattar

I en ond kungs fängelsehålor sitter några logiker. Den onde kungen bestämmer sig en dag för att sätta dem på prov: Han säger till dem att han tänker ställa dem på ett rakt led och ge dem var sin hatt. Hatten är antingen svart eller vit, men de får inte se färgen själva. I ledet kommer de alla kunna se de logiker (och deras hattar) som står framför dem, men inte de som står bakom. Logikerna kommer sedan i tur och ordning få frågan vilken färg det är på deras hatt, och när de är klara kommer alla som gissade fel att bli avrättade. Den som står sist i ledet kommer att tillfrågas först, sedan den näst sista, och så vidare tills alla tillfrågats. Logikerna tillåts inte kommunicera alls med varandra, annat än att de hör vad de andra gissar.

Hänsynslöst kan det tyckas, men kungen är åtminstone snäll nog att låta logikerna prata ihop sig först och komma överens om en strategi. Hur ska de gå till väga för att *garantera* att så många som möjligt överlever?

Ledtråd

De kan förstås omöjligt garantera att den första tillfrågade logikern (han som står sist i ledet) överlever, eftersom all information han har är de andras hattar. Men vilken information kan de se till att den nästa i ledet har, och alla efter honom? Kan de rentav se till att *bara* den första logikern riskerar att dö?

Lösning

Lösning: De kan garantera att bara en logiker riskerar att dö, nämligen den som tillfrågas först. Vi kommer kalla honom Räknaren, och hädanefter kallar vi endast resten av logikerna för *ledet*. Räknaren ser förstås alla i ledet, eftersom han står längst bak.

Logikernas strategi bör vara att låta Räknaren räkna antalet svarta hattar framför honom i ledet, och ifall det är ett udda antal hattar säger han ”Svart”, annars säger han ”Vit”.

Antag nu till exempel att Räknaren säger ”Svart”. Nästa logiker vet då att ifall inte han också ser ett udda antal svarta hattar framför sig i ledet, så måste hans egen hatt vara svart och säger då detta. Ifall han ser ett udda antal svarta hattar, då kan inte hans egen hatt vara det, och därmed säger han ”Vit”.

För varje logiker i ledet gäller alltså:

1) Han vet ifall det finns ett udda eller jämnt antal svarta hattar i ledet, för att Räknaren berättat detta.

2) Han vet färgen på alla hattar bakom honom i ledet, för att han har hört vad de sagt.

3) Han ser färgen på alla hattar framför honom i ledet.

Om han känner till färgen på alla hattar i ledet utom sin egen, och dessutom vet om antalet svarta hattar i ledet är udda eller jämnt, då kan han enkelt lista ut sin egen hattfärg.

Den första logikern, Räknaren, överlever bara om han har tur och färgen han använder för att koda udda eller jämnt råkar överensstämma med hans egen hattfärg. Detta är oundvikligt, eftersom hans enda information är de övrigas hattar, och dessas färg är oberoende av hans egen, och kan alltså omöjligt ge honom någon information som hjälper honom.

Diskussion: Lösningen visar sig faktiskt vara mycket användbar i datorsammanhang, och används för att få säkrare lagring på hårddiskar. Metoden kallas RAID (*Redundant Array of Independent Disks*) och låter dig lagra information på ett godtyckligt antal hårddiskar, och med hjälp av bara en extra hårddisk kunna garantera att om någon hårddisk går sönder så förlorar man ändå ingen information!

Låt säga att vi har lagrat de binära strängarna 11010001, 00011011 och 10100010 på tre olika hårddiskar. Vi använder då en extra hårddisk som Räknare, och bygger upp en binärsträng R som uppfyller att ifall de tre strängarna har ett udda antal 1:or på plats k så har R en 1:a på den platsen, annars en 0:a. I vårt exempel får vi:

$$
\begin{array}{r l}
\text{Hårddisk } A\text{:} & 1\ 1\ 0\ 1\ 0\ 0\ 0\ 1 \\
\text{Hårddisk } B\text{:} & 0\ 0\ 0\ 1\ 1\ 0\ 1\ 1 \\
\text{Hårddisk } C\text{:} & 1\ 0\ 1\ 0\ 0\ 0\ 1\ 0 \\
\hline
\text{Hårddisk } R\text{:} & 0\ 1\ 1\ 0\ 1\ 0\ 0\ 0
\end{array}
$$

Antag nu att till exempel hårddisk B går sönder. Då kan vi återskapa dess innehåll från de övriga, eftersom vi vet från R att på första platsen på A, B och C fanns totalt ett jämnt antal 1:or. Eftersom A och C tillsammans har ett jämnt antal 1:or på den platsen, så måste B ha haft en 0:a där. På samma sätt finner vi B:s värde på alla positioner, och har räddat innehållet.

Detta är mycket behändigare än att ha säkerhetskopior på alla hårddiskar, eftersom vi då skulle behöva $2n$ hårddiskar för att lagra n hårddiskars information säkert. Med RAID-metoden behöver vi bara $n + 1$ hårddiskar för att vara garanterade att varje hårddisk går att återskapa från de andra. Om däremot mer än en hårddisk går sönder på samma gång, då kan informationen från dessa inte räddas.

Logikernas lösning är bara ett specialfall av RAID-metoden, där varje logiker är en hårddisk som kan lagra endast ett binärt tal (som kan ha värdet svart eller vit). Varje logiker kan ”återskapa” färgen på sin hatt genom att kolla vilken färg de andra logikerna har och jämföra detta mot Räknaren. Att använda RAID på riktiga hårddiskar motsvarar att logikerna har kanske åtskilliga miljarder hattar på huvudet (och tvingas gissa färgen på varje hatt för sig), och att varje logiker (utom Räknaren) ändå garanterat kan lista ut färgen på var och en av sina hattar!