Obligatorisk oppgave nr 1 i INF110

Formalia

Hver student leverer sin egen løsning av oppgaven til gruppelærereren på den gruppen som hun eller han er oppmeldt på.

Det som skal leveres, er utskrift av fyldig kommentert kildekode i Java og filnavn på en kjørbar versjon av programmet. Kildekodefilen skal ikke være lesbar for andre enn deg, mens selve programmet skal være kjørbart for alle. Svar per e-post vil ikke bli godkjent.

Innleveringsfrist: Fredag 14. september 2001

Dronningoppgaven

Gitt et "sjakkbrett" på $n \times n$ ruter. Oppgaven består i å plassere n dronninger på dette slik at ingen kan slå noen annen. Etter sjakkens regler er det slik at en dronning kan slå en annen dersom de to dronningene står i samme horisontale linje, samme vertikale linje, eller samme diagonal.

Tenker man seg en løsning på dette problemet vil syv "andre" (ikke alle nødvendigvis forskjellige) løsninger framkomme ved å anvende følgende transformasjoner:

- Speiling om horisontal og om vertikal midtlinje.
- Speiling om hoveddiagonal og om bidiagonal.
- Rotasjon (f.eks. *med* klokka) 90, 180 og 270 grader.

Denne gruppen av åtte løsninger (medregnet den vi startet med) er da slik at om vi starter med en tilfeldig valgt av dem, så vil nettopp denne gruppen framkomme ved å anvende de syv transformasjonene. Dermed kan mengden av løsninger på oppgaven over deles opp i grupper (matematikerne ville kalle dem "ekvivalensklasser"), innen hvilke løsningene bare er rotasjoner og speilinger av hverandre. Mellom løsninger i forskjellige grupper er det aldri slik.

Du skal skrive et program som, for en gitt verdi av n, skriver ut en og bare en representant for hver av de nevnte løsningsgruppene.

Programmet skrives i Java.

Oppgaven kan løses elegant med rekursiv generering av permutasjoner med innlagt "avskjæring". Man kan da passelig gå ut fra den algoritmen for å generere alle permutasjoner for en gitt n som er gjennomgått på forelesningene. Hovedavskjæringen går ut på at man ikke fortsetter med permutasjonsgenerering dersom den løsningen vi nå har i $p[0..i]$ umulig kan

bygges ut til en brukbar løsning på dronningoppgaven, ved at den f.eks. allerede har indre kollisjoner i seg.

Det finnes flere måter å sørge for at man bare får skrevet ut én løsning i hver symmetrigruppe. En måte er å lagre de løsningene vi skriver ut og så sammenligne vår løsning med disse, men dette krever mye tid og plass. Da er det mer elegant å bygge på det faktum at vi generer permutasjonene (og dermed løsningene) i "stigende" rekkefølge, og sørge for bare å skrive ut den første vi kommer til i hver gruppe, som derved også er den minste i hver gruppe. Problemet blir da hvordan vi for en gitt løsning kan teste om den er den første i sin gruppe. Erfaringsmessig synes mange at dette er den vanskeligste delen av oppgaven. Tenk gjennom problemet. Det er satt av tid på gruppene slik at du kan få diskutert ditt forslag til løsning med andre studenter og gruppelæreren din.

Det finnes mange angrepsvinkler på denne oppgaven, og selv om vi anbefaler den som er antydet over, må du gjerne forsøke med andre. For å få den godkjent bør det da imidlertid være noe snertent med løsningen, og den bør ikke ta (særlig) lengre tid enn en standardløsning ut fra forslaget over. La imidlertid ikke dette skremme deg fra å forsøke andre muligheter – det er mye å lære av det.

Løsningen antydet over kan også gjøres mer eller mindre fiks, noe som i stor grad kan innvirke på tiden beregningene tar. Det gjelder stadig å finne ut så tidlig som mulig om den delvise løsningen man sitter med kan bygges ut til en fullstendig løsning eller ikke.

Du kan drive tiden litt ekstra ned ved å fjerne de rekursive kallene, og i stedet "simulere" dette på annen måte. Dette vil vi imidlertid ikke anbefale, da det gjerne ødelegger mye av programstrukturen, samt at noe av vitsen med oppgaven er at du får trening i rekursiv tankegang.