Datastrukturer og Algoritmer, dat C
Henning Christiansen
14/10-2004

Opgaver til forelæsningen 14/10-2003 om Rekursion, dynamisk programmering, m.v.

Det foreslås at prioritere 1–3.1 højest. Opgave 4-5 er opgaver som henvender sig til specielt interesserede (og løsninger vil ikke blive gennemgået med mindre der er generel stemning derfor).

Opgave 1
Fortsætter en diskussion fra forelæsningen omkring sortering ved del-og-hersk. Den såkaldte flettesortering (merge sort) — og det er ikke nødvendigt at læse fremad i bogen. Opgaven her er selvforklarende.

Det handler om at sortere spillekort, og vi antager følgende ordningsrelation:

- Klør er mindre end hjerter, som er mindre end spar, som er mindre end ruder.
- Hvis to kort har samme farve, så bestemmes rækkefølge af værdien, dvs. 1 (Es) er mindre end 2, som ...., som er mindre end 12, som er mindre end 13 (Konge).

Udfør som rollespil en del-og-hersk sortering af et blandet spil kort:

- En bunke med 1 kort er sorteret; Ellers:
- Bunken deles i to ca. lige store delbunker.
- Hver bunke sorteres (af hver sin hjælper)
- De to sorterede bunker flettes ved at man lægger dem med »ansigtet« opad og til enhver tid tager det midste af de to kort man kan se og lægger over i en ny bunke.

En af de studerende på holdet starter, og han eller hun kan delegere sortering af delbunker ud til andre studerende (forudsat de er ledige), som vil udføre samme algoritme.

Vurdér tidskompleksiteten ved følgende scenarier:

1. Vi har en kraftig parallel-computer som til enhver tid har en ledig processor til at overtage en delopgave (svarende at vi havde haft studerende nok på holdet).
2. Vi har en normal sekventiel computer, svarende til at en og samme studerende må fungere som sine egne hjælpere (og deres hjælpere og hjælperes hjælpere).

Det er OK at argumentere intuitivt; ingen grund til at benytte bogens generelle formel.

Opgave 2
Betragt de fire versioner af Fibonacci-beregnere i foreslæsningsOHerne (via kursets hjemmeside), og angive "O" udtryk for dem.

1. Angiv "O" udtryk for dem hver især, når de kaldes én enkelt gang med et eller anden *n*.
2. Angive "O" udtryk for et program som ikke laver andet end at kalde på Fibonacci-metoden. Det antages, at den kaldes *m* gange med et argument som er begrænset opadtil med *n*.

Opgave 3
Betragt programmet side 271 i bogen som handler om at give byttepenge tilbage (programtekst med hovedprogram kan findes via link på kursushjemmesiden).

1. I forelæsningen forklarede vi kun den del af algoritmen, som handlede om at bestemme det mindste antal antal mønter, som skal til for at give et vist beløb tilbage. Men algoritmen indeholder faktisk også noget ekstra, som gør det mulig at udskrive hvilke mønter der faktisk er tale om. Men det er ikke forklaret hvordan eller hvorfor det virker — og de benyttede variabelnavne er heller ikke hjælpsomme. Opgaven lyder nu: Find ud af og forklar for hinanden hvordan denne del af algoritmen virker.
2. Nu skal programmet tilpasses danske forhold, og for nemheds skyld, regn alt i ørebeløb. Indsæt data svarende til danske sedler og mønter, og tilføj princippet "rund-op-ned-til-nærmeste-25øre". Implementér og aftest programmet.

Opgave 4 (Hjemmearbejde til folk med god tid)
Tag udgangspunkt i programmet "drawRuler" side 247 i bogen (programtekst med hovedprogram han findes via link på kursushjemmesiden). Det indeholder noget primitivt grafik, som viser hvordan man definere sig en tegneplade og tegne en streg. Det er sådan set det eneste vi skal bruge fra det program. Brug disse faciliteter til at løse så mange af følgende opgaver som du gider, eller hit selv på nogen, der er sjovere:

- Skriv et program som tegner en ramme – og rekursivt inde i den en ramme som er 85% mindre på begge leder — og rekursivt inde i den .... . Vær opmærksom på, du selv må hitte på et stopkriterium.
- Du har sikkert læst "ramme" som et rektangel hvis sider er parallelle med tegnefladens sider. Skriv nu et program som først tegner en ramme (i nævnte betydning). Dernæst inde i den en rombe (dvs. en firkant på højkant), hvis hjørner rører ved rammens sider. Inde i den igen en ramme hvis hjørner rører ved rombens sider — og så fremdeles. Benyt to metoder som skiftevis kalder hinanden.
- Tegn en rekursiv busk, som ca. ser ud som et "Y" med to mindre buske i grenene. I første omgang, så Y'erne alle peger opad, men prøv at dreje dem lidt hver gang, så det ser mere dekorativt ud.
- Bland Fibonacci-tal ind i historien og se om du kan få tegningen til at konvertere mod det gyldne snit. (Det gylde snit kan findes som grænseværedien for forholdet mellem to Fibonacci-tal:

  1/1, 1/2, 2/3, 3/5, 5/8, 8/13, 13/21, 21/34...

Opgave 5 (Hjemmearbejde til folk med ekstra meget tid)
Fortsættelse af opgave 3.
Der findes faktisk mere optimale måder at give penge tilbage på end den som vor algoritme når frem til. Tag beløbet 99kr. Med danske mønter/sedler vil vil forvente at det eksisterende program vil give 50+20+20+5+2+2, dvs. 6 sedler/mønter. Men det kan gøres med to enheder: "har du en krone, får du en hund af mig". Tilpas algoritmen så den tillader denne måde at give tilbage på og minimer det totale antal mønter/sedler, som gives frem og tilbage.
NB. Opgaven er svær: Læreren har ikke selv kunnet løse den.