Institut for Matematik og Datalogi
Syddansk Universitet, Odense

30. januar 2020
RF

DM507 – Opgaver uge 6

I denne uge er timerne med opgaveregning (også kaldet e-timer, eksaminatorier eller øvelsestimer) undtagelsesvis uden forberedelse, dvs. du arbejder med opgaverne i klassen. Resten af året vil e-timerne være *med* forberedelse, dvs. der gennemgås opgaver, som du har løst (eller forsøgt at løse) *før* timerne.

Det er vigtigt for dit udbytte af kurset at du resten af året *reelt forsøger at løse opgaverne* før e-timerne – alene og/eller sammen med andre (f.eks. i studiegrupper). Afsæt et antal minutter til at tænke over hver opgave. Forhåbentligt når du så langt, at du får skrevet en løsningen ned til hovedparten af dem. Lidt svære opgaver vil være mærket med $(*)$, og mere svære opgaver vil være mærket med $(**)$

I denne uge skal du medbringe laptop med Java eller Python installeret til e-timerne.

Eksaminatorier

1. Cormen et al. problem 1.1 (side 14). Erstat dog "microseconds" med "nanoseconds" (dvs. 10^{-9} sekunder), da dette ca. er hvad en CPU-cyklus tager på en moderne processor. Du skal kun udfylde søjlerne *second*, *day*, *century*. For nogle af indgangene kan man finde svaret ved matematisk udregning, for andre må man prøve sig frem ved at indsætte forskellige værdier af n.

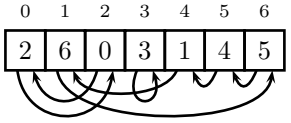
2. Brodals noter om puslespil, opgave 1.

3. Brodals noter om puslespil, opgave 2. Her betyder "optimal følge af ombytninger" et antal ombytninger som angivet af sætning 1 i noterne. Opgaven viser, at andre algoritmer end "grådige algoritmer" (algoritmer som altid bringer mindst ét element på plads) kan være optimale for dette puslespilsproblem.

4. Lav et Java- eller Python-program, som genererer en tilfældig permutation af heltallene fra 0 til $n-1$ (for et n som er en input parameter). I Java kan man bruge typen `ArrayList` samt metoden `shuffle` fra `Collections` utility klassen. I Python kan man bruge lister samt funktionen `shuffle` fra modulet `random`. Udskriv tallene i din permutation.

5. ($*$) Hvis et array/en liste indeholder en permutation af tallene 0 til $n-1$ kan man definere kredse på samme måde som for puslespillet fra første forelæsning: et tal x, som står på plads y i arrayet, giver en pil fra plads y til plads x (dvs. hvis tallet 1 står på plads 4 i arrayet, er der en pil fra plads 4 til plads 1), og en samling pile, der hænger sammen i en cyklisk kæde, kaldes en kreds.

   Her er et eksempel, hvor der i alt er tre kredse i permutationen (check at du finde dem):



   Lav en algoritme, som tæller antal kredse i en permutation. Hvad er køretiden for din algoritme som funktion af n?

   Implementer din algoritme i Java eller Python. Kør dit program mange gange på en tilfældig permutation (se opgaven ovenfor) og gem output fra alle kørsler.

   Brug data fra disse eksperimenter til at give et bud på sandsynligheden for, at der i en tilfældig permutation med $n = 16$ er k kredse, for $k = 1, 2, \ldots, 16$. Dvs. lav din egen version af figur 3 i noterne, men med n lig 16 og ikke 64.

   Find også det gennemsnitlige antal kredse i dine eksperimenter. Passer dit tal med formlen på side 4 i noterne, dvs. er det tæt på $H_{16} = \sum_{i=1}^{16} 1/i = 1/1 + 1/2 + 1/3 + \ldots + 1/16$?