

Algoritmer (datastrukturer) och komplexitet
våren 2009

Mästarprov 1: Algoritmer

Mästarprovet ska lösas **individuellt** och redovisas både skriftligt och muntligt. *Inget samarbete är tillåtet, se vidare hederskodexen.*

Skriftliga lösningar ska lämnas senast **måndag 2 mars klockan 10.15** på föreläsningen eller senast klockan 10.00 samma dag i kursens inlämningslåda på studentexpeditionen på Osquars backe 2, plan 2. Det är viktigt att du lämnar in i tid! Skriv ditt namn och personnummer överst på framsidan. Se till att spara en kopia av dina lösningar så att du kan läsa på inför den **muntliga redovisningen** som kommer att ske senare samma vecka för någon av lärarna. Boka tid för en femton minuters muntlig redovisning med kommandot `bok new adk09`. Redovisningstider kommer att läggas upp senast 26 februari. Gör helst bokningen före inlämningsdagen 2 mars. Du kan alltid avboka din bokning.

Mästarprovet är ett obligatoriskt och betygsatt moment i kursen. Det består av tre uppgifter som motsvarar betygsriterierna för E, C respektive A. För godkänt (betyg E) krävs helt rätt på en av uppgifterna. Helt rätt på två av uppgifterna ger betyg C och alla rätt ger betyg A. Ett mindre fel på en uppgift sänker betyget ett steg. Läs mer om betygsriterier och slutbetyg i kurs-PM eller på kursens webbsida.

1. SL-upplysningen

Betygsriterium: utveckla algoritmer med datastrukturer för enkla problem givet en konstruktionsmetod.

På `sl.se` kan man skriva in två hållplatser i Storstockholm, ange en önskad starttid och få reda på hur man snabbast kommer från starthållplatsen till sluthållplatsen med SL, dvs genom att byta mellan olika buss-, tåg- och tunnelbanelinjer. I denna uppgift ska du konstruera en effektiv algoritm som löser det problemet. Du kan anta att algoritmen har tillgång till en databas med alla tidtabeller för alla hållplatser. Databasen tar emot ett anrop bestående av en hållplats h och en tidpunkt t och skickar tillbaka en lista med alla hållplatser som man kan åka till från h och hur många minuter efter t man kommer fram till varje hållplats. Alla tal är heltal.

Anta att det finns n hållplatser i Storstockholm och att bytestiden (kortaste tiden från att man stiger av ett transportmedel till att man stiger på nästa) vid varje hållplats är 1 minut. Din algoritm behöver bara mata ut hur många minuter den kortaste resan är, inte vilka byten man ska göra.

Beskriv algoritmen med pseudokod och analysera algoritmens tidskomplexitet uttryckt i n . Motivera på en hög nivå (motsatsen till detaljnivå) att algoritmen hittar den kortaste restiden.

Ledning: Formulera problemet som ett grafproblem och använd en känd algoritm från kursen för att lösa det. Tänk på att kantvikterna inte är kända från början.

Vänd!

2. Energisnålt garage för tåg

Betygskriterium: utveckla algoritmer med datastrukturer för icketriviala problem.

SJ ska bygga ett nytt stort garage för sina tåg. Experter på hållbar utveckling säger att byggnaden bör göras kvadratisk för att material- och uppvärmningskostnader ska minimeras. Naturligtvis ska byggnaden inte heller göras större än nödvändigt. Din uppgift är att konstruera en algoritm som givet en beskrivning av alla tågsätt som ska få plats i garaget talar om vilken storlek M det garaget ska ha (det vill säga längden på kvadratens sidor).

Indata är n stycken positiva heltal t_1, \dots, t_n där t_i anger längden på tågsätt nr i . Längden anges i tågbredder. Varje tågsätt är mellan 1 och n tågbredder långt. Om garaget är M långt så består det av M stycken parallella spår av längden M . Tågen anländer i tur och ordning (från nr 1 till n) till garaget och parkerar efter varandra på de M spåren i garaget enligt följande giriga regel:

När tåg i kommer fram till garaget växlar det in på det spår som det finns minst plats kvar på, men ändå $\geq t_i$ plats kvar på, och kör fram tills det snuddar vid framförvarande tåg. Om det inte finns tillräckligt med plats på något spår som det redan parkerats tåg på så parkerar tåget längst fram på det vänstraste oanvända spåret.

Vilket är det minsta M som gör att alla tåg får plats i garaget om dom parkerar med ovanstående regel?

Du ska också analysera tidskomplexiteten för din algoritm och ge ett resonemang som visar att algoritmen är korrekt. Algoritmen ska gå i $O(n \log^2 n)$ med enhetskostnad för att ge helt rätt. En algoritm som inte klarar den gränsen men som går i $O(n^2)$ räknas som ett *mindre fel*.

3. Hållbart fiske

Betygskriterium: utveckla algoritmer med datastrukturer för svårare problem med den metod som passar bäst.

Experter i hållbar utveckling har utformat ett ganska komplicerat kvotsystem för fiske som bygger på *fiskebiljetter*. Varje fiskare får k stycken fiskebiljetter numrerade 1 till k och som ska användas i nummerordning. Varje gång fiskaren fiskar gör han eller hon av med en fiskebiljett. Hur mycket fisk som fiskaren får dra upp då beror på för vilken gång i ordningen fiskaren fiskar, alltså fiskebiljettens nummer. Man får bara fiska en gång vid varje fiskeplats.

Anta att en fiskare åker en rutt som i tur och ordning passerar n stycken lämpliga fiskeplatser, numrerade 1 och n , och som vill fiska vid k av dessa fiskeplatser. Fiskaren vill veta vilka k platser hon ska fiska vid för att få mest fisk.

Din uppgift är att konstruera en effektiv algoritm som givet fiskekvoterna $k_{i,j}$ (som står för hur många fiskar man får ta upp vid fiskeplats i om man använder fiskebiljett j), där $1 \leq i \leq n$ och $1 \leq j \leq k$, hittar och skriver ut vilka k fiskeplatser som ska användas i den bästa lösningen. Om flera lösningar är precis lika sannolika så räcker det att algoritmen ger en av dom.

Du ska också analysera tidskomplexiteten för din algoritm och ge ett övertygande resonemang som visar att den är korrekt. Algoritmen ska vara så effektiv som möjligt ($O(nk)$ är tillräckligt för godkänt).