

DD1352 Algoritmer, datastrukturer och komplexitet, hösten 2016

Mästarprov 1: Algoritmer

Mästarprovet ska lösas **individuellt** och redovisas både skriftligt och muntligt. *Inget samarbete är tillåtet, se vidare hederskodexen.* Du ska alltså inte diskutera lösningar med någon annan fram till dess att alla muntliga redovisningar är avklarade.

Skriftliga lösningar ska lämnas senast **måndag 10 oktober klockan 15.15** på föreläsningen eller senast klockan 15.00 samma dag i kursens inlämningsfack vid receptionen, Osquars backe 2, plan 4. Det är viktigt att du lämnar in i tid!

Skriv ditt namn och personnummer överst på framsidan av lösningarna. Se till att spara en kopia av dina lösningar så att du kan läsa på inför den **muntliga redovisningen** som kommer att ske 14–20 oktober för någon av lärarna. Boka tid för en femton minuters muntlig redovisning på kurswebbsidan senast 10 oktober klockan 15. Bokningslistorna läggs upp senast 6 oktober. Om du inte hinner göra uppgiften så avbokar du enkelt din bokning.

Det är viktigt att du förbereder dig inför den muntliga redovisningen. För att en uppgift ska godkännas ska du kunna förklara och motivera algoritmen muntligt och reda ut eventuella oklarheter.

Läs uppgifterna mycket noga så att du inte råkar basera dina lösningar på en missuppfattning. Fråga en lärare på kursen om något är oklart.

Mästarprovet är ett obligatoriskt 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 betyg på kurswebben.

För att se exempel på hur utförliga lösningarna bör vara kan du titta på lösningar till *tidigare mästarprov* på kurswebben. Där finns också tips om hur man skriver pseudokod, som du bör läsa innan du formulerar dina algoritmer. Noggrannare kriterier för hur bedömningen går finns på kurswebben.

1. Uppgiftsutdelning

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

En samvetsgrann lärare på KTH vill dela ut uppgifter till sina n studenter så att inga studenter som känner varandra får samma uppgift. Läraren, som är optimistisk och lite lat, tror att det räcker att hitta på två uppgifter. Skriv en algoritm som testar om det stämmer. Indata är en lista med par av studenter som känner varandra. Anta i denna uppgift att vänskap är ömsesidigt.

Modellera problemet med en graf som har kanter mellan de hörn som representerar studenter som känner varandra. Algoritmen ska baseras på en metodisk genomgång av grafen med BFS eller DFS.

Beskriv algoritmen med pseudokod, förklara varför den fungerar och analysera dess tidskomplexitet, som ska vara polynomisk.

2. Optimal Pokémonvandring med jaktfri ruvning

Betygskriterium: utveckla algoritmer med datastrukturer för icke-triviala problem.

I mobilspelet Pokémon Go finns det två sätt att samla nya Pokémon: dels kan man fånga vilda Pokémon, som brukar dyka upp i närheten av så kallade Pokéstopp (som är platser i verkligheten), dels kan man kläcka Pokémon ur ägg som ruvas i särskilda ägginkubatorer medan man promenerar. Det finns tre sorters ägg: ägg som kläcks efter 2 kilometers promenad, ägg som kläcks efter 5 kilometers promenad och ägg som kläcks efter 10 kilometers promenad.

I denna uppgift ska du hjälpa en Pokémon Go-spelare, som vi kan kalla Arrietty, att planera sin vandring så att hon kan samla så många Pokémon som möjligt. Arrietty har n_2 stycken 2-kilometersägg, n_5 stycken 5-kilometersägg och n_{10} stycken 10-kilometersägg. Hon har sett ut en vandring som är $2n_2 + 5n_5 + 10n_{10}$ kilometer lång, så att hon precis ska kunna kläcka alla ägg (hon har bara tillgång till en ägginkubator, så hon kan bara ruva ett ägg i taget under vandringen). Arrietty har gjort en lista över dom Pokéstopp där hon kan behöva stanna för att ta hand om en nykläckt Pokémon och börja ruva på ett nytt ägg. Listan har punkter för 2 km in i vandringen, 4 km in i vandringen, 5 km in i vandringen osv. Antalet Pokéstopp som man kan nå efter i kilometer beskrivs av ett icke-negativt heltal $P[i]$ i en array $P[2..len]$ där $len = 2n_2 + 5n_5 + 10n_{10}$. Arrietty är fokuserad på vandringen och tänker bara stanna dom $n_2 + n_5 + n_{10}$ gånger som krävs för att avsluta en ruvning och påbörja en ny. Hon kommer inte att försöka fånga några Pokémon medan hon går, utan bara när hon stannat.

Problemet är att givet n_2, n_5, n_{10} och $P[2..len]$ maximera antalet Pokéstopp som Arrietty når dom $n_2 + n_5 + n_{10}$ gånger hon stannar under vandringen, så att hon ska kunna fånga så många Pokémon som möjligt. Vandringsvägen är bestämd i förväg, så det enda som kan varieras är i vilken ordning äggen ska ruvas.

Definiera lämpliga delproblem och utveckla en effektiv (polynomisk) algoritm som löser dessa delproblem och med deras hjälp löser Arriettys problem. Det räcker att hitta det maximala *antalet* Pokéstopp som kan nås; algoritmen behöver inte konstruera vilken äggordning som ger detta maximala antal.

Beskriv algoritmen med pseudokod och analysera dess tidskomplexitet uttryckt i n_2, n_5 och n_{10} . Motivera också att algoritmen är korrekt.

3. Boka klassrum optimalt

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

En skola har ont om klassrum och har gett dig i uppdrag att konstruera en algoritm som sköter skolans rumsbokning på bästa tänkbara sätt. I skolan finns k identiska klassrum. Bokningen av klassrummen görs terminsvis efter (det mycket oregelbundna) lektionsschemat. Varje schemalagd lektion anges med en starttid s_i och en sluttid f_i . För enkelhets skull antar vi att alla tider anges som heltal. Hur dessa relaterar till verkliga dagar och klockslag är inte intressant i denna uppgift. Naturligtvis kan bara en lektion bokas i varje klassrum vid varje tidpunkt. Konstruera en effektiv algoritm som gör en optimal klassrumsbokning, det vill säga hittar en rumsbokning i dom tillgängliga k klassrummen som gör att så många lektioner som möjligt tilldelas ett klassrum. (För återstående lektioner får rektorn hyra andra lokaler.) Indata beskrivs av talet k samt n stycken schemalagda lektioner, givna som talpar (s_i, f_i) . Alla tal i indata är positiva heltal. Utdata ska vara en lista för varje klassrum med vilka lektioner som i tidsordning ska bokas i det rummet. Du kan vidare anta att $k < n$, för annars är problemet trivialt.

Beskriv algoritmen med pseudokod och analysera tidskomplexiteten. För godkänt krävs en algoritm som har optimal tidskomplexitet (uttryckt i n). Visa därför att övre och undre gräns sammanfaller. En i övrigt helt korrekt lösning med en algoritm som är en log-faktor ifrån undre gränsen ger sänkning med ett betygssteg.

Du ska motivera att din algoritm är korrekt.