DD1352 Algoritmer, datastrukturer och komplexitet, hösten 2016

Mästarprov 2: Komplexitet

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 **6 december 2016 klockan 12.15** på föreläsningen eller senast klockan 12.00 samma dag i kursens inlämningslåda inne på 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 9–15 december. Boka tid för en femton minuters muntlig redovisning på kurswebbsidan senast 6 december klockan 12. Bokningslistorna läggs upp senast 1 december. 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 lösningen 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.

Mästarprov 2 är ett obligatoriskt moment i kursen. Det består av tre uppgifter som motsvarar betygskriterierna för E, C respektive A. Helt rätt på uppgift 1 eller 2 ger betyg E. Helt rätt på uppgift 1 *och* 2 ger betyg C. Helt rätt på alla uppgifter 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 en vägledning till hur man genomför och formulerar bevis. Noggrannare kriterier för bedömningen av den skrifliga och muntliga redovisningen finns på kurswebben.

1. **Ultrasmart maraton**

   Betygskriterium: förklara principerna, utföra enklare reduktioner mellan givna problem.

   Ultrasmart maraton är en sport som kräver både uthållighet och slughet. Banan består av ett antal vägskäl och vägar som förbinder dem. Vägarna kan gå i tunnlar under och på broar över andra vägar. Vägarna kan också gå i kringelikrokar, så en väg mellan två vägskäl kan vara mycket längre än fågelvägen. Därför kan indata representeras som en oriktad graf där vägskälen är hörn och kantvikterna, som kan vara godtyckliga heltal, anger väglängder.

   Det finns ingen särskild startpunkt; man väljer själv vid vilket vägskäl man vill starta. Man väljer också var man vill gå i mål. Däremot måste man besöka varje vägskäl precis en gång. Den som lyckas springa det längsta loppet har vunnit.

   I ultrasmart maraton delas det också ut ett specialpris till de löpare som, utöver att fullfölja loppet, har sprungit minst hälften så långt som längden av alla vägar som ingår i banan.

   Visa att det är NP-fullständigt att avgöra om det är möjligt att vinna ett specialpris. Visa att problemet är NP-svårt genom att reducera problemet Hamiltonsk stig (se övning 9 för definition).

2. **Schemaproblemet**
   Betygskriterium: visa NP-fullständighet.

   En skola (samma skola som i sista uppgiften i mästarprov 1, men det gör inte denna uppgift enklare att lösa) tycker att det är för dyrt att hyra lokaler på stan, så man har nu bestämt att all undervisning ska bokas i dom k tillgängliga klassrummen (som numreras från 1 till k). Skolan vill skaffa ett system som kan schemalägga lektionerna på lämpliga tider och i lämpliga klassrum.

   För varje lektion anger skolan hur lång den ska vara och eventuella andra krav, som kan vara av tre typer: att den ska schemaläggas i en viss sal (t ex så att en musiklektion läggs i musiksalen), att den inte får schemaläggas överlappande i tid med vissa andra lektioner (för att det är samma lärare eller samma elever), eller att den ska schemaläggas efter en viss annan lektion (så att den startar vid en senare tidpunkt än den andra lektionen slutar, för att det ska gå att få lektioner i en önskad ordning).

   Tider anges som ickenegativa heltal. Hur det kopplar till verkliga klockslag är inte relevant i denna uppgift.

   Vid varje tidpunkt kan naturligtvis bara en lektion pågå i varje lektionssal. Om en lektion avslutas vid tidpunkt t kan nästa lektion i samma sal börja tidigast vid samma tid t.

   Skolan vill optimera så att den lektion som slutar sist slutar så tidigt som möjligt.

   Du ska nu övertyga skolan om att problemet är för svårt för att lösa optimalt genom att formulera problemet som ett beslutsproblem och visa att detta beslutsproblem är NP-fullständigt.

   Indata till problemet är antalet klassrum k och n stycken lektioner. För varje lektion anges dess längd och eventuella ytterligare krav (av dom tre typerna ovan). Beslutsproblemet har dessutom ett mål *timelimit* som indata.

   I reduktionen får du som känt NP-fullständigt problem använda problem som visats eller påståtts vara NP-fullständigt på föreläsningarna, i föreläsningsanteckningarna eller i kursboken.

3. **Konstruktion av schema**
   Betygskriterium: göra konstruktionsreduktioner.

   Åter till schemaproblemet i uppgift 2.

   Anta att det finns en algoritm ExistsSchedule($k, lesson[1..n], timelimit$) som löser beslutsproblemet i tid $B(n, timelimit)$, som är en positiv växande funktion (som förmodligen växer exponentiellt i n och polynomiskt i antalet bitar i *timelimit*).

   Konstruera en polynomisk turingreduktion av konstruktionsproblemet till beslutsproblemet, dvs konstruera en algoritm Schedule($k, lesson[1..n]$) som med hjälp av anrop till ExistsSchedule konstruerar och skriver ut ett optimalt schema (vilka lektioner som ska schemaläggas när och i vilken sal). Beskriv reduktionen med pseudokod.

   Även om det finns flera möjliga optimala lösningar så ska bara en skrivas ut.

   Analysera tidskomplexiteten för din reduktion och motivera att den är korrekt.