Viggo Kann och Stefan Nilsson, KTH CSC

Lösningar till teoritenta i Algoritmer, datastrukturer och komplexitet 2016-12-19

1. (6 p) Är följande påståenden sanna eller falska? För varje deluppgift ger riktigt svar 1 poäng och ett *övertygande motiverat* riktigt svar 2 poäng.

   a) $3^n \in \omega(2^n)$.
   Sant. $\lim_{n\to\infty} \frac{3^n}{2^n} = \lim_{n\to\infty}(3/2)^n = \infty$, vilket innebär att $3^n \in \omega(2^n)$.

   b) En Lasvegasalgoritm är en probabilistisk algoritm som går i polynomisk tid.
   Falskt. En Lasvegasalgoritm är visserligen en probabilistisk algoritm, men exekveringstiden behöver inte vara polynomisk.

   c) Mästarsatsen lämpar sig för att beräkna tidskomplexiteten för dynamiskprogrammeringsalgoritmer.
   Falskt. Dynamiskprogrammeringsalgoritmer implementerar en rekursion genom att alla värden beräknas iterativt från basfallen och uppåt. Mästarsatsen kan bara användas för att beräkna rekursioner med en speciell struktur som uppkommer för dekompositionsalgoritmer.

2. (3 p) A, B, C och D är beslutsproblem. Anta att B är NP-fullständigt och att det finns polynomiska Karpreduktioner mellan problemen så här (en reduktion av A till B tecknas här $A \to B$):

$$\begin{array}{ccc} A & \leftarrow & B \\ \downarrow & & \uparrow \\ C & \leftarrow & D \end{array}$$

   Vad vet man då om komplexiteten för A, C och D? Sätt ett kryss i tabellen nedan för det man säkert vet och en ring för det som är möjligt men som man inte vet säkert.

	ligger i NP	är NP-fullständigt	är NP-svårt
A	○	○	X
C	○	○	X
D	X	○	○

3. (2 p) a) Vad är den engelska termen för *totalsökning*?

   Exhaustive search.

   b) Vad är den svenska termen för *computability*?

   Beräkningsbarhet.

4. (3 p; 1 p på a och 2 p på b)
   a) Definiera begreppet *oavgörbart problem.*

   Ett beslutsproblem som inte kan lösas i ändlig tid av någon algoritm.

   b) Definiera komplexitetsklassen *NP.* (Ge bara en definition.)

   Mängden av beslutsproblem som har en verifieringsalgoritm, som kan verifiera ja-lösningar i polynomisk tid. Det betyder att verifieringsalgoritmen, som tar instansen och en lösningssträng som indata, svarar nej för varje nej-instans, oavsett lösningssträng, och att det för varje ja-instans finns en polynomiskt lång lösningssträng så att verifieringsalgoritmen svarar ja.

5. (Uppgift för betyg D)
   Ge två konceptuellt olika förslag på hur man kan angripa NP-svåra beslutsproblem som inte är förtäckta optimeringsproblem (som vanligt förutsatt att P≠NP).

   1. Gör en totalsökningsalgoritm som kan lösa problemet för små indata.

   2. Begränsa problemet genom att lägga på restriktioner som är rimliga att göra.

   En annan möjlighet är att försöka hitta en probabilistisk algoritm för problemet.

6. (Uppgift för betyg C)
   I *Kappsäcksproblemet* är indata ett tal W och n prylar, där pryl i har vikten w_i och värdet v_i. Problemet är att bestämma vilka prylar som ska packas ner i kappsäcken så att summan av deras vikt är högst W och summan av deras värde är maximal.

   Vi söker i denna uppgift en heuristik som försöker hitta en bra lösning till Kappsäcksproblemet, ju större sammanlagt värde desto bättre. Heuristiken ska använda sig av *lokal sökning*. Beskriv din algoritm med pseudokod på relativt hög nivå.

Extend(K, W, E) = // utvidgar kappsäcken K med saker i E till viktgränsen W
    $sw \leftarrow 0; sv \leftarrow 0$
    foreach $(w, v) \in K$ **do** $sw \leftarrow sw + w; sv \leftarrow sv + v$
    foreach $(w, v) \in E$ **do** // behandla elementen i E i slumpvis ordning
        if $s + w \leq W$ **then**
            $K \leftarrow K \cup \{(w, v)\}; sw \leftarrow sw + w; sv \leftarrow sv + v$
    return (K, sv)

KnapsackLocalSearch($\{(w_i, v_i)\}_1^n, W$) =
    $(K, s) \leftarrow$ Extend($\emptyset, W, \{(w_i, v_i)\}_1^n$)
    $R \leftarrow \{(w_i, v_i)\}_1^n - K$ // resterande element
    $oldval \leftarrow 0$
    while $s > oldval$ **do**
        foreach $(w, v) \in K$ **do** // prova att byta ut ett element i kappsäcken
            $(newK, news) \leftarrow$ Extend($K - \{(w, v)\}, W, R$)
            if $news > s$ **then** // bättre lösning funnen
                $K \leftarrow newK; s \leftarrow news$
                $R \leftarrow \{(w_i, v_i)\}_1^n - K$
    return (K, s)

Alla kursregistrerade får inom kort en kursenkät som var och en uppmanas att svara på så snart som möjligt. Stefan, Viggo och 2017 års elever på kursen tackar på förhand!

Vill du ha högre betyg på kursen? Om du efter (om)mästarproven har fått minst betyg C på två av dom betygsatta kursmomenten (teoritentan, mästarprov 1, mästarprov 2) och minst betyg E på det tredje så får du boka in dej på muntlig redovisning 11-13 januari 2017, se kurswebben.
Om du blir godkänd på teoritentan och redovisar extralabben 10 januari så får du räkna extralabbsbetyget som teoritentabetyg.
Det går bra att i senare kursomgångar plussa mästarproven.
Ommästarproven läggs idag upp på kurswebben. Redovisning är i början av januari.