

Algoritmer (datastrukturer) och komplexitet våren 2010

Ommästarprov 1: Algoritmer

Detta mästarprov är avsett för den som ännu inte är godkänd på mästarprov 1. Det är bara en uppgift och den kan bara ge betyg E. Mästarprovet ska lösas **individuellt** och redovisas både skriftligt och muntligt. Inget samarbete är tillåtet, se vidare hederskodexen.

Du ska lämna in den **skriftliga lösningen** på studerandeexpeditionen (Osquars backe 2, plan 2) senast klockan 14.00 den 24 maj 2010. Den **muntliga redovisningen** är 26 maj 2010. Boka tidigast 20 maj och senast 24 maj en tid för muntlig redovisning på kurswebbsidan.

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 uppgiften mycket noga så att du inte råkar basera dina lösningar på en missuppfattning. Fråga Viggo om något är oklart.

Japanskt bildpussel

Ett japanskt bildpussel är ett fritidsfördriv där det gäller att fylla i vissa av rutorna i ett kvadratisk rutmönster så att en bild bildas. Till vänster om varje rad och ovanför varje kolumn står en rad med heltal. Talen talar om hur stora grupper av ifyllda rutor som ska finnas på raden/kolumnen. Om det till exempel står **2 5 1** betyder det att det ska finnas tre grupper, där den första (sett från vänster till höger/uppifrån och ner) består av två ifyllda rutor, nästa av fem ifyllda rutor och den tredje av en enda ifylld ruta. Grupperna måste skiljas åt av en eller flera tomma (alltså oifyllda) rutor.

Bildpusslen utformas alltid så att det finns en unik lösning, och om man tittar på lösningen på avstånd ser man att dom ifyllda rutorna skapar en bild. Se ett löst exempel på andra sidan.

Din uppgift är att konstruera och analysera två olika totalsökningsalgoritmer som löser ett japanskt bildpussel i ett $n \times n$ -rutmönster. Indata är två arrayer *Rowspec*[1 : n] och *Colspec*[1 : n] med talföljder. Varje talföljd innehåller högst 10 tal (positiva heltal) som alltså beskriver hur långa grupperna av ifyllda rutor är i motsvarande rad/kolumn.

Problemet är faktiskt NP-fullständigt, så dina algoritmer kommer att vara exponentiella. Den första algoritmen ska ha tidskomplexitet $O(2^{n^2} \cdot n^2)$ och den andra algoritmen ska ha tidskomplexitet $O(n^{10n+2})$. Algoritmerna ska utföra totalsökningen på (principiellt) helt olika sätt. Beskriv algoritmerna med pseudokod; hög nivå på pseudokoden går bra så länge den går bra att tyda. Beräkna sedan vilken av algoritmerna som har lägst tidskomplexitet.

