

**Lösningar till teoritenta i Algoritmer (datastrukturer) och komplexitet
för KTH DD1352–2354 och SU 2007-12-15**

1. A) **Falskt.** $\frac{n!}{2^n} = \frac{n(n-1)\dots 2 \cdot 1}{2 \cdot 2 \dots 2} \geq \frac{3 \cdot 3 \dots 3}{2 \cdot 2 \dots 2} 1 \frac{1}{2} = \frac{1}{2} \left(\frac{3}{2}\right)^{n-2}$.
 - B) **Falskt.** Det som krävs av en Monte Carlo-algoritm är att sannolikheten för ett korrekt Ja-svar skall vara $> \frac{1}{2}$. Något sådant krav finns inte för vanliga ickedeterministiska algoritmer.
 - C) **Falskt.** Problemet är avgörbart. Vi kan köra beräkningen $M(x)$ och har den inte stannat efter k steg så avbryter vi och svarar Nej.
 - D) **Sant.** Skissartat kan det inses på följande sätt: Om ett problem ligger i NP så finns det en ickedeterministisk algoritm som avgör problemet. Den har typiskt exponentiellt många beräkningsvägar och att undersöka dem alla tar exponentiell tid. Men de kräver bara polynomiellt minnesutrymme (för att hålla reda på vilken beräkningsväg man är på.) Beräkningarna kan därför utföras med en turingmaskin med polynomiell minnesåtgång.
2. Vad som krävs här är att man nämner att en approximationsalgoritm i *polynomiell tid* hittar ett approximativt värde på lösningen till ett optimeringsproblem. Approximationskvoten är definierad som $\frac{OPT}{APP}$ för maximeringsproblem och $\frac{APP}{OPT}$ för minimeringsproblem. För beskrivning av approximationsalgoritmen för HÖRNTÄCKNING se föreläsningssanteckningarna.
 3. a) Om vi har en instans G av HAMILTONCYKEL kan vi skapa en instans av TSP genom att skapa en komplett graf G' med samma noder som G . Sätt kantvikterna till 1 om kanten finns med i G och sätt den till 2 annars. Sätt sedan $K = n$ d.v.s. ställ frågan om det finns väg runt G' av längd n .
b) Vi vet att båda problemen är NP-fullständiga. Så vi vet t.ex. att $SAT \leq HAMILTONCYKEL$. Enligt Cooks sats är SAT NP-fullständigt. Alltså gäller $TSP \leq SAT$. Kombinerar vi detta får vi $TSP \leq HAMILTONCYKEL$.
 4. Algoritmen är FLOYD-Warshalls algoritm. Arrayen d innehåller kortaste avståndet mellan alla par av noder i grafen G . Det är ett exempel på en dynamisk programmeringsalgoritm.
 5. En graf G är tvåfärgbar om och endast om den är *bipartit*. Det kan avgöras med en variant av BFS-sökning. Vi antar för enkelhets skull att grafen är sammanhängande. Välj en nod som startnod och beräna sedan avståndet från den till alla noder. De noder som har jämnt avstånd färgas sedan röda och de övriga blå. Kontrollera om detta ger en korrekt färgning. Om den är korrekt är grafen 2-färgbar, annars inte. BFS och kontrollen av färgningen går i polynomiell tid.