

**Lösningar till teoritentan i Algoritmer, datastrukturer och komplexitet
2019-12-17**

1. (4 p) Är följande påståenden sanna eller falska? För varje deluppgift ger riktigt svar 1 poäng och ett *övertygande motiverat* riktigt svar 2 poäng.

a) $2^n \in \Omega(e^n/n)$.

Falskt. Påståendet säger att vänsterledet växer minst lika snabbt som e^n/n asymptotiskt. Vi måste därför kontrollera gränsvärdet för kvoten, för om det är 0 är påståendet falskt.

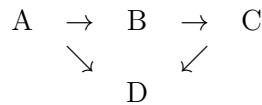
$$\lim_{n \rightarrow \infty} \frac{2^n}{e^n/n} = \lim_{n \rightarrow \infty} \frac{n}{(e/2)^n} = 0$$

eftersom $e/2 > 1$ och n växer långsammare än exponentialfunktionen.

b) En lista med n stycken personnummer ska sorteras. Radixsortering är då effektivare än heapsortering.

Sant. Personnummer har 10 siffror. Med radixsortering tar det tid $O(n)$ att sortera tal med konstant antal siffror. Heapsortering gör $O(n \log n)$ jämförelser av personnummer, och varje jämförelse tar konstant tid. Radixsortering är därför effektivare.

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



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) Anta att X är ett optimeringsproblem (taget från verkligheten) som man skulle vilja lösa, men som man efter modellering (matematisk formulering) visat är NP-svårt. Förklara varför det är värdefullt att både angripa problemet med en approximationsalgoritm och med heuristiker.

En approximationsalgoritm ger en garanti för hur långt från den optimala lösningen den producerade lösningen är. Om någon av heuristikerna ger ett bättre värde än approximationsalgoritmen så vet man att det också uppfyller samma garanti.

4. (2 p; 1 p för a och 1 p för b)

a) Vad är den engelska termen för *beslutsproblem*?

Decision problem.

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

Beräkningsbarhet.

5. (3 p; 1 p för varje definition)

Komplexitetsklassen P består av alla beslutsproblem som kan lösas med en algoritm som går i polynomisk tid i värsta fallet.

a) Definiera begreppet *beslutsproblem*.

Ett problem som formuleras som en fråga som har svaret ja eller nej.

b) Definiera begreppet *algoritm*.

En ändlig beskrivning av hur man steg för steg löser ett problem.

c) Definiera begreppet *polynomisk tid*.

Antalet steg begränsas av $O(n^k)$ för någon konstant k , där n är indatas längd.