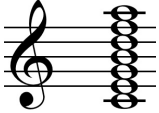


Måndagen den 15 oktober 2012 kl 9–13

Maxpoäng = 100. 50 poäng ger E, men den som fått 47-49 poäng kan få komplettera. Gränserna för högre betyg är 60, 70, 80, och 90 poäng. Skriv upp antal bonuspoäng från labbar respektive hemtal på tentaomslaget.

Hjälpmedel: En algoritmbok och ditt handskrivna formelblad.

1. *Torn i musiken.*
(8p) Konstruera och rita upp en KMP-automat som söker efter TONTORN.
Ange även next-vektorn!

2. *Datastrukturen Torn*
(16p) Ett Torn är en sorterad stack, med största elementet i botten och det minsta överst. Insättningsmetoden `put(x)` ska lägga in ett element på rätt plats i tornet. Man kan implementera `put` med hjälp av två abstrakta stackar!
 - a. Rita ett exempel som visar hur detta kan göras när man stoppar in talen 8 3 5 9 i tornet.
 - b. Beskriv algoritmen! Du måste använda stackarnas operationer `push`, `pop` och `isEmpty`.
 - c. Ett alternativ till tornet är en min-heap. Visa hur talen ovan skulle sättas in i en min-heap.
 - d. Vilken datastruktur är effektivast - tornet eller heapen? Jämför komplexiteten för både insättning och utplockning av element.
3. *Högsta tornet*
(16p) Du vill göra en lista över alla torn i världen som är högre än hundra meter. Till din hjälp har du data från 1000 städer: för varje stad finns en lista med data (bland annat höjden) för stadens 100 000 byggnader (ordnade efter kvarter). Det tar fyra minuter att i tur och ordning söka igenom alla städernas byggnader.
 - a. Hur lång tid skulle det ta om man genomförde sökningarna parallellt (alla städer samtidigt) på en flerkärning dator (med Go) istället för sekventiellt?
 - b. Att sortera husen (utan parallellprogrammering) i en enda av städerna efter höjd tar 10 sekunder med urvalssortering (selection sort). Hur lång tid tar det att sortera alla 1000 städernas hus med samma algoritm?
 - c. Att sortera husen i en enda av städerna efter höjd tar en halvsekund med mergesort. Hur lång tid tar det att sortera alla 1000 städernas hus med samma algoritm?
 - d. Skulle vi ha tjänat på att istället börja med att slå ihop alla städerna och sortera med mergesort? Motivera ditt svar!
4. *Krypterat torn*
(4p) Kryptera LEKTORNS LIKTORN med transpositionschiffer!

5. *Teori*

(20p) Nedan finns fem frågor om algoritmer och datastrukturer. Varje fråga kan ge upp till fyra poäng. Motivering krävs!

- Ordna följande funktioner efter hur snabbt dom växer:
 $n, 2^n, n^2, \log(n), 512n^2, n^3$
- Vad är skillnaden mellan huffmankodning och LZW?
- Vad ska man tänka på när man skapar en hashfunktion?
- Visa med ett exempel hur en regel i en BNF-syntax kan se ut, och förklara hur den ska tolkas.
- Vad är ett pivot-värde i Quicksort? Förklara och visa ett exempel.

6. *Schackturn i binärträd*

(12p) I ett unikt schackparti förekom åtta Torn på brädet.

- Sortera i tur och ordning in följande positioner i ett binärt sökträd (bokstavsordning): b2 e5 f6 g7 d4 c3 h8 a1.
- Vad blir utskriften när trädet skrivs ut i preorder, inorder och postorder?
- Hur många jämförelser måste man göra som mest när man söker i ditt träd? Rita upp det optimala sökträdet, och visa i vilken ordning värdena skulle ha lagts in för att vi skulle få det trädet.

7. *Hissen i tornet*

(12p) Laget Taipei Assassains (som vann VM i League of Legends igår) har rest hem för att fira segern. Hissen i Tornet i Taipei (101 våningar, 509 meter högt) har bara två knappar: UPP (upp u våningar) och NER (ner n våningar).



- Visa hur man ska göra för att ta sig från våning 90 till våning 99 om UPP åker upp 10 våningar och NER åker ner 7.
- Vad är indata och utdata till det generella problemet?
- Beskriv en algoritm för ett program som löser problemet!

8. *Potenstorn*

(14p) Ett potenstorn är ett matematiskt uttryck på formen $a_1^{a_2^{\dots^{a_n}}}$.

- Berätta vilken typ av variabel du lagrar indata i.
- Potenstornet ska tolkas så här: $a_1^{(a_2^{\dots^{a_n}})}$. Formulera en *rekursiv* tanke (med basfall) för en funktion som beräknar potenstornets värde.
- Att skriva ett program som jämför två potenstorn med varandra visar sig vara lättare sagt än gjort. Varför det?