

TDDI16 Datastrukturer och algoritmer
Tentamen (TEN1)
2015-01-08, 8–12

Examinator: Tommy Färnqvist
Jour: Tommy Färnqvist (telefon 013-282479).
Max poäng: 22 poäng (betyg 5 = 19p, 4 = 15p, 3 = 11p)
Hjälpmedel: INGA HJÄLPMEDEL TILLÅTNA!!!

VÄNLIGEN IAKTTAG FÖLJANDE

- Lösningar till olika problem skall placeras enkelsidigt på separata blad. Skriv inte två lösningar på samma papper.
- Sortera lösningarna innan de lämnas in.
- MOTIVERA DINA SVAR ORDENTLIGT: avsaknad av, eller otillräckliga, förklaringar resulterar i poängavdrag. Även felaktiga svar kan ge poäng om de är korrekt motiverade.
- Om ett problem medger flera olika lösningar, t.ex. algoritmer med olika tidskomplexitet, ger endast optimala lösningar maximalt antal poäng.
- SE TILL ATT DINA LÖSNINGAR/SVAR ÄR LÄSBARA.
- Lämna plats för kommentarer.

Lycka till!

1. Vilka av följande påståenden är sanna och vilka är falska? Svar utan motivering ger inga poäng. (2 p)

(a) $f(n) \in O(h(n))$ och $g(n) \in O(h(n)) \Rightarrow f(n) + g(n) \in O(h(n))$ (1)

(b) $\Theta(n^2) \subset \Theta(n^3)$ (1)

2. Betänk en kryptografisk metod där ett meddelande representeras av ett stort primtal p . Meddelandet krypteras mha ett annat stort primtal $q > p$, så att meddelandet som faktiskt skickas över nätverket är $r = p \cdot q$. Oturligt nog fångas ditt meddelande upp av någon som använder följande dekrypteringsalgoritm: (4 p)

För varje heltal p , sådant att $1 < p < r$, kontrollera om p delar r . Om det är fallet, stanna och returnera p som det dekrypterade meddelandet, annars fortsätt.

- (a) Vad är värstafallstidskomplexiteten för dekrypteringsalgoritmen? Eftersom indata r lagras som en bitsträng, låt indatastorleken n vara antalet byte som behövs för att lagra r , dvs $n = \log_2(r)/8$, och antag att varje division tar tid $O(n)$. (2)

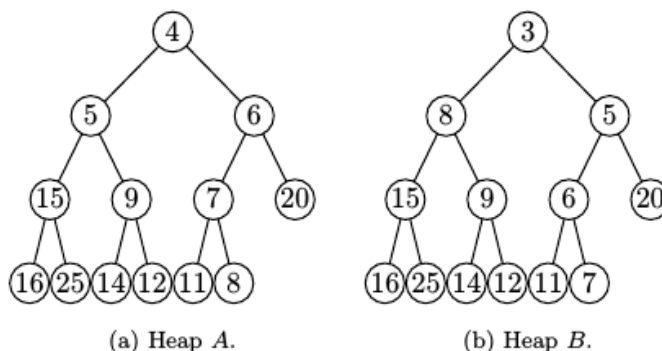
- (b) Antag att varje division av två heltal med upp till 100 bitar vardera kan utföras på 1 mikrosekund (en miljondels sekund). Hur lång tid, i värsta fallet, tar det att dekryptera ett meddelande innehållande 100 bitar med algoritmen ovan? Borde du oroa dig för säkerheten den kryptografiska metoden ger dig? (2)

3. Antag att vi låter en klass AVL-träd tillåta flera noder med samma nyckel. Dvs, sökträdesegenskapen blir att för varje nod v , som lagrar (k_v, x_v) , i ett AVL-träd T gäller: (4 p)

- Nycklar lagrade i v 's vänstra äkta delträd är mindre än eller lika med k .
- Nycklar lagrade i v 's högra äkta delträd är större än eller lika med k .

Beskriv (med pseudokod) en metod `findAllElements(k)` i AVL-trädet. Algoritmen ska ha tidskomplexitet $O(\log(n)+s)$, där n är trädets storlek och s är antalet element som returneras (dvs antalet noder i trädet vars nyckel är k). Motivera tidskomplexiteten för din algoritm.

4. Vilken minimal sekvens av operationerna `insert` och/eller `removeMin` utförda på heap A transformerar den till heap B ? Rita heapen efter varje operation. (3 p)



5. Rita en representation av en hashtabell H med 11 celler och dess innehåll efter att vi använt avbildningen (3 p)

$$h(k) = 3k + 5 \pmod{11}$$

för att sätta in nycklarna 10, 11, 20, 15, 40, 9, 16, 22 i den initialt tomma H . Antag att vi hanterar kollisioner med separat länkning (*separate chaining*).

6. Låt S vara en sekvens av n element på vilken en total ordningsrelation är definierad. En *inversion* i S är ett par av element x och y sådant att x står före y i S men $x > y$ gäller. (6 p)
- (a) Beskriv (med pseudokod) en algoritm med exekveringstid $O(n \log(n))$ som beräknar antalet inversioner i S . Motivera algoritmens korrekthet noggrant. (4)
- (b) Analysera algoritmens tidskomplexitet i värsta fallet. (2)
- (Tips: försök modifiera algoritmen för merge-sort.)