

OMTENTAMEN I
DATASTRUKTURER OCH ALGORITMER DAV B03

130819 kl. 08:15 – 13:15

Ansvarig Lärare: Donald F. Ross

Hjälpmedel: Bilaga A: Algoritmerna.

***** OBS *****

Ni som har läst från och med HT 2006

Betygsgräns:

Kurs: Max 60p, Med beröm godkänd 50p, Icke utan beröm godkänd 40p, Godkänd 30p
(varav **minimum 15p från tentamen, 15p från labbarna**)
Tentamen: Max 30p, betyg 5: 26p-30p, betyg 4: 21p-25p, betyg 3: 15p-20p
Labbarna: Max 30p, betyg 5: 26p-30p, betyg 4: 21p-25p, betyg 3: 15p-20p

Ni som har läst tidigare än HT 2006

Betygsgräns:

Kurs: Max 60p, Med beröm godkänd 50p, Icke utan beröm godkänd 40p, Godkänd 30p
(varav **minimum 20p från tentamen, 10p från labbarna**)
Tentamen: Max 40p, betyg 5: 34p-40p, betyg 4: 27p-33p, betyg 3: 20p-26p
Labbarna: Max 20p, betyg 5: 18p-20p, betyg 4: 14p-17p, betyg 3: 10p-13p

SKRIV TYDLIGT – LÄS UPPGIFTERNA NOGGRANT

***** OBS *** Denna tentamen är kopierad på båda sidor *** OBS *****

(1) Ge ett kortfattat svar till följande uppgifter.

- (a) Vad är ett "free tree"?
- (b) Ge **ett exempel** av ett "free tree".
- (c) Ge **ett exempel** av hur man kan använda en riktade graf i verkligheten.
- (d) I en graf, vad representerar kanterna?
- (e) Vad är nackdelen med hashning?
- (f) Vad är "kvadratisk probning" inom hashning?
- (g) I stället för att mäta tid för sorterings algoritmer som vi har gjort i labb 2, föreslå ett annat sätt att mäta prestandet.
- (h) Varför är begreppet "samling" (collection) viktigt inom datastrukturer och algoritmer?
- (i) Vad är skillnaden mellan begreppen "sorterad" och "ordnad"?
- (j) Vad är fördelen med binära träd jämfört med generella träd?

Totalt 5p**(2) Sekvens**

Diskutera ingående följande påstående "**sekvensen är den viktigaste ADT:en inom datavetenskap**".

5p**(3) Träd**

Hur fungerar balansering i ett AVL-träd? . **Diskutera ingående.**
Ange gärna konkreta exempel. Ange alla antagande.

5p**(4) Övriga frågor – Dijkstra kontra Prim**

Se Bilaga A för algoritmerna.

- (a) Förklara principerna bakom **Dijkstras algorithm.**
- (b) Förklara principerna bakom **Prims algorithm.**
- (c) Jämför **Dijkstras algoritm** med **Prims algoritm.**

2p

2p

1p

Totalt 5p

(5) Labbkod

- (a) I graflabben har en student skrivit följande kod för att lägga till en kant (edge) från an adjacency lista. Förklara **ingående** hur koden fungerar. Använd gärna exempel. **Ange alla antagande.**

Vilka är förutsättningarna för att koden ska fungera?

```
void adde(char cs, char cd, int v) {  
    set_edges(b_findn(cs, G), b_adde(cd, v, get_edges(b_findn(cs, G))));  
}
```

2p

- (b) **Beskriv ingående** hur du skulle **testa** träd labben (BST samt komplett träd). Vilka operationer har man på ett träd och hur skulle Du testa varje operation?

3p

Totalt 5p

(6) Graf - Prims algorithm

Se Bilaga A för algoritmerna.

Tillämpa **den givna Prims algoritm nedan** på **den oriktade grafen**,

(a-6-b, a-3-c, a-7-d, b-1-c, b-12-e, c-4-d, c-5-e, c-9-f, d-3-f, e-8-f).

Börja med nod "a".

Ange *alla* antaganden och visa *alla* beräkningar och mellanresultat

(3p)

Rita en bild av varje steg under algoritmens exekvering för att förklara processen.

(2p)

Totalt 5p

Bilaga A: Algoritmerna

```

1) Prim ( node v ) -- v is the start node
2) {
3)   U = {v}; for i in (V-U) { low-cost[i] = C[v,i]; closest[i] = v; }

4)   while (!is_empty (V-U) ) {
5)     i = first(V-U); min = low-cost[i]; k = i;
6)     for j in (V-U-k) if (low-cost[j] < min) {min = low-cost[j]; k = j; }
7)     display(k, closest[k]);
8)     U = U + k
9)     for j in (V-U) if ( C[k,j] < low-cost[j] ) ) {low-cost[j] = C[k,j]; closest[j] = k; }
10)  }
11) }

```

```

1)   Dijkstra ( a ) -- a is the start node
2)   {
3)     S = {a}
4)     for (i in V-S) D[i] = C[a, i]      -- initialise D
5)     for (i in 1..(|V|-1)) {
6)       choose w in V-S such that D[w] is a minimum
7)       S = S + {w}
8)       foreach ( v in V-S ) D[v] = min(D[v], D[w]+C[w,v])
9)     }
10)  }

```