Skriftlig Eksamen
Algoritmer og Datastrukturer (DM507)

Institut for Matematik og Datalogi
Syddansk Universitet, Odense

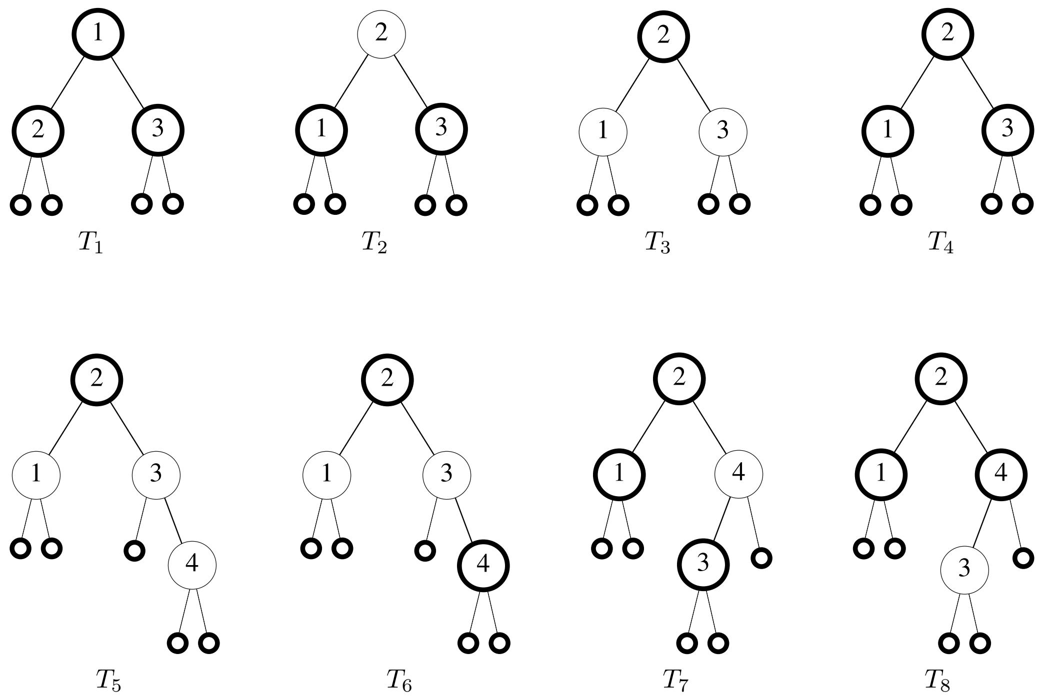
Tirsdag den 14. juni 2011, kl. 9:00–13:00

Alle sædvanlige hjælpemidler (lærebøger, notater, osv.) samt brug af lommeregner er tilladt.

Eksamenssættet består af 5 opgaver på 10 nummererede sider (1–10).
Fuld besvarelse er besvarelse af alle 5 opgaver.
De enkelte opgavers vægt ved bedømmelsen er angivet i procent.

Der må gerne refereres til algoritmer og resultater fra lærebogen (Cormen et al. Introduction to Algorithms. 3. udgave) inklusive øvelsesopgaverne. Henvisninger til andre bøger accepteres ikke som besvarelse af et spørgsmål.

Husk at begrunde dine svar!



Figur 1: De otte træer fra spørgsmål 1a

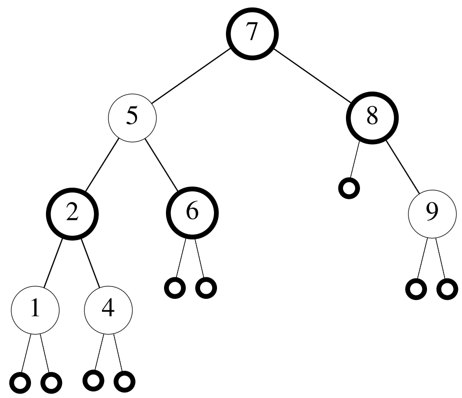
Opgave 1 (18%)

Denne opgave handler om rød-sorte træer.

Spørgsmål a (8%):

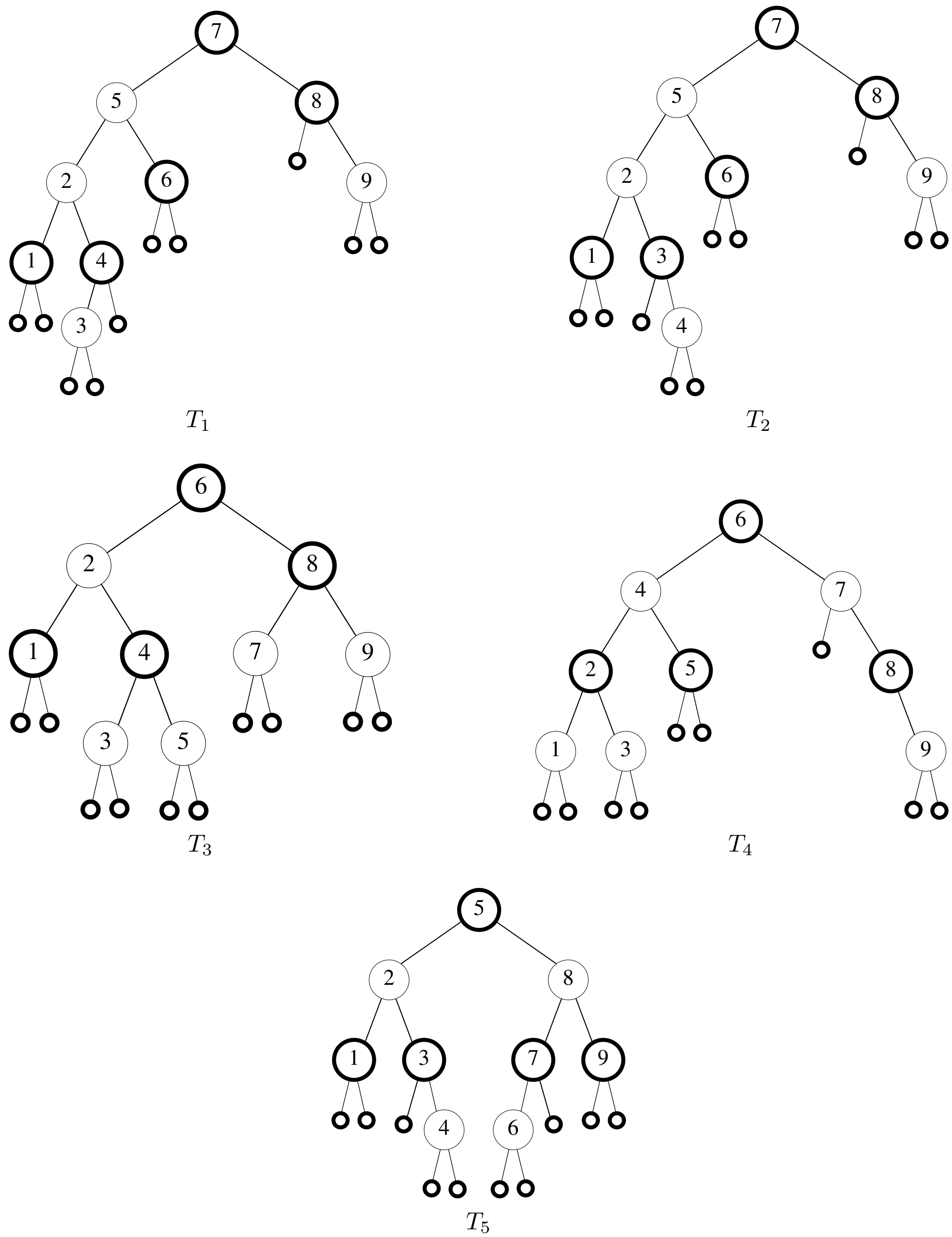
Betragt de otte træer i figur 1. Knuder tegnet med fed er sorte, og resten er røde. Angiv, hvilke af de otte træer der er rød-sorte træer. ☐

Spørgsmål b (10%): Betragt nedenstående rød-sorte træ T.

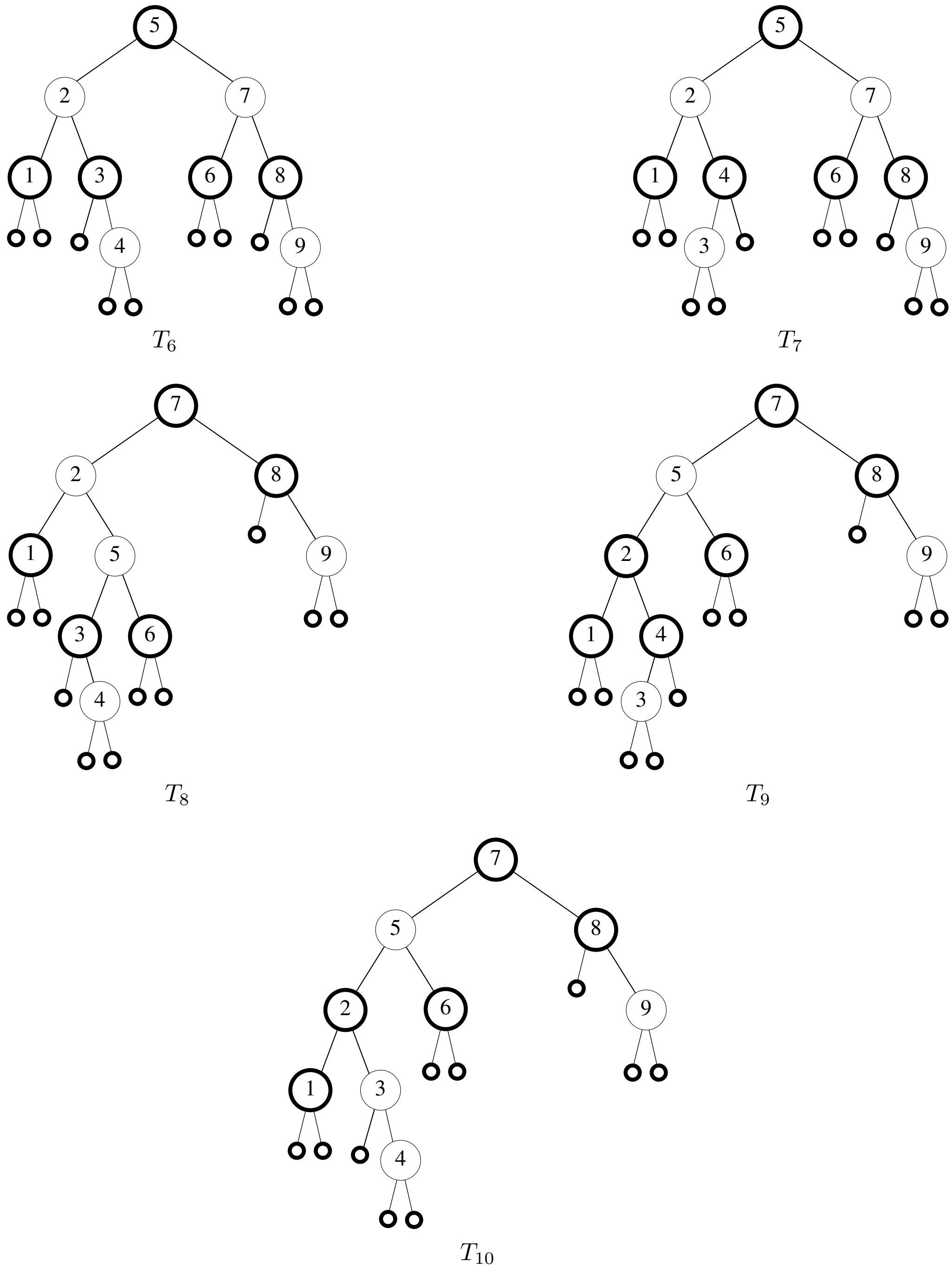


Nu indsættes en knude med nøgle 3.

Hvilket af træerne i figur 2 og 3 svarer til T, efter at knuden med nøgle 3 er indsat? ☐



Figur 2: De første fem træer fra spørgsmål 1b



Figur 3: De sidste fem træer fra spørgsmål 1b

Opgave 2 (15%)

Denne opgave handler om asymptotisk notation.

Lad f_1, f_2, g_1 og g_2 være positive funktioner, som opfylder, at

- $f_1(n) \in O(g_1(n))$
- $f_2(n) \in O(g_2(n))$

Hvilke af følgende tre udsagn er da sande?

(a) $f_1(n) + f_2(n) \in O\big(g_1(n) + g_2(n)\big)$

(b) $g_1(n) + g_2(n) \in \Omega\big(f_1(n) + f_2(n)\big)$

(c) $\dfrac{f_1(n)}{f_2(n)} \in O\left(\dfrac{g_1(n)}{g_2(n)}\right)$

Opgave 3 (24%)

I denne opgave ser vi på sortering i to special-tilfælde:

1. Alle nøgler er forskellige, og input er omvendt sorteret; d.v.s. nøglerne optræder i faldende orden. Eks: $\{12, 9, 8, 5, 2\}$.
2. Alle nøgler er ens. Eks: $\{5, 5, 5, 5, 5\}$.

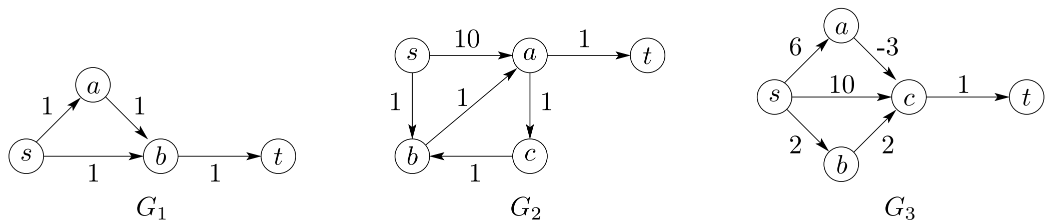
I begge tilfælde antages det, at alle nøgler er heltal mellem 0 og n^5, hvor n er antallet af nøgler, der skal sorteres.

Spørgsmål a (6%): Angiv køretiden for Insertion Sort i hvert af de to tilfælde. ☐

Spørgsmål b (6%): Angiv køretiden for Quicksort i hvert af de to tilfælde. ☐

Spørgsmål c (6%): Angiv køretiden for Heapsort i hvert af de to tilfælde. ☐

Spørgsmål d (6%): Angiv den bedst mulige køretid for Radix Sort i hvert af de to tilfælde. ☐



Figur 4: Tre vægtede grafer

Opgave 4 (18%)

I denne opgave ser vi på tre forskellige algoritmer anvendt på hver af de tre vægtede grafer i figur 4. Som sædvanligt er længden af en vej i en vægtet graf summen af vægtene på kanterne i vejen.

Spørgsmål a (6%): For hvilke af de tre grafer i figur 4 kan man bruge Bredde-Først-Søgning til at beregne den korteste afstand fra s til t? ☐

Spørgsmål b (6%): For hvilke af de tre grafer i figur 4 kan man bruge DAG-SHORTEST-PATHS til at beregne den korteste afstand fra s til t? ☐

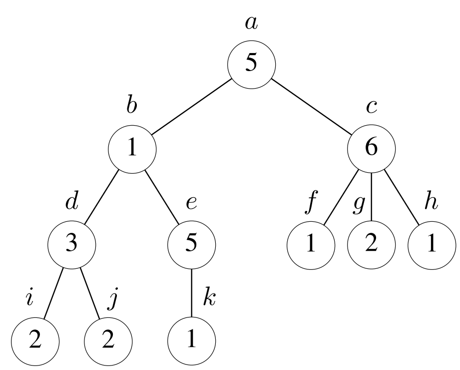
Spørgsmål c (6%): For hvilke af de tre grafer i figur 4 kan man bruge Dijkstras Algoritme til at beregne den korteste afstand fra s til t? ☐

Opgave 5 (25%)

I denne opgave er der givet et træ T, som har en rod. Hver knude i træet har en vægt, som er et positivt heltal. Figur 5a viser et eksempel på et sådant træ.

En delmængde af knuderne i T er *uafhængig*, hvis den ikke indeholder nogen naboknuder. D.v.s. hvis en uafhænging mængde indeholder en knude x, da indeholder den ingen af x's børn og heller ikke x's forælder.

Opgaven går ud på at finde en uafhængig mængde med størst mulig samlet vægt.



(a) Et vægtet træ T_1 med rod

x	a	b	c	d	e	f	g	h	i	j	k
Med x	18	6	6	3	5	1	2	1	2	2	1
Uden x	15	9	4	4	1	0	0	0	0	0	0

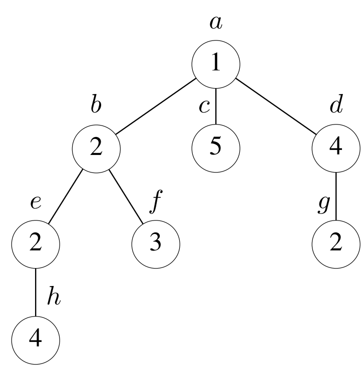
(b) Optimale delløsninger med og uden roden i undertræer af T_1

Figur 5: Et eksempel

Eksempel:

Figur 5a viser et vægtet træ T_1 med rod a. Knuderne a, e, f, g, h, i og j udgør en uafhængig mængde med samlet vægt 18. Denne mængde er en optimal løsning, da ingen andre uafhængige delmængder af knuderne i T_1 har større samlet vægt.

For hver knude x i T_1 angiver tabellen i figur 5b den optimale vægt for undertræet med rod i x:
Første række angiver den størst mulige vægt, hvis x *skal* være med i den uafhængige mængde, og anden række angiver den størst mulige vægt, hvis x *ikke* må være med.



Figur 6: Et vægtet træ T_2 med rod

Spørgsmål a (10%): Udfyld nedenstående tabel for træet T_2 vist i figur 6.

x	a	b	c	d	e	f	g	h
Med x								
Uden x								

□

Spørgsmål b (5%): Angiv en uafhængig mængde med størst mulig vægt i T_2. □

Spørgsmål c (10%): Angiv en algoritme baseret på dynamisk programmering, som givet et vægtet træ med rod finder den størst mulige vægt af en uafhængig mængde i træet. Algoritmen behøver ikke finde den tilsvarende uafhængige mængde, blot dens vægt.

Hvad er din algoritmes køretid og pladsforbrug? □