

Aufgabe 1

Das folgende Speicherabbild enthält den Kopf (*head*) einer einfach verketteten Liste an der Adresse 27. Ein (Listen)Knoten besteht aus zwei benachbarten Speicherzellen. Die erste Zelle enthält einen hexadezimalen Datenwert und der zweite eine hexadezimale Adresse.

Gib die Werte in dieser einfach verketteten Liste in der richtigen Reihenfolge an.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	X	4A	21	3D	17	C2	00	31	2C	37	3B	00	25	35	00	3E
1	00	05	38	3D	0C	00	10	23	00	2C	14	02	6A	1F	3B	06
2	14	3A	09	00	00	07	A5	1A	11	0C	30	3E	29	00	19	00
3	0E	00	15	28	22	15	26	B9	48	10	00	0D	0F	2D	1F	23

- Das erste Adress-Halbbyte bezeichnet die Zeile, das zweite die Spalte.
- 00 steht für den Nullzeiger und ist keine gültige Adresse.
- Der Speicher kann verwaiste Daten enthalten.

Aufgabe 2

Das obere Speicherabbild enthält den Kopf (*head*) einer einfach verketteten Liste an der Adresse 1D. Ein (Listen)Knoten besteht aus zwei benachbarten Speicherzellen. Die erste Zelle enthält einen hexadezimalen Datenwert und der zweite eine hexadezimale Adresse.

In diese Datenstruktur soll effizient der Wert 2F am Listenanfang eingefügt werden.

Stelle die *veränderten* Werte im Speicherabbild *nach* dem Einfügen des Knotens im unteren Raster dar. Verwende für den neuen Knoten einen beliebigen freien Speicherbereich.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	X	25	3E				25	00			11	14		24	01	
1					2C	00	38	2B	31	0A				18	06	

- Das erste Adress-Halbbyte bezeichnet die Zeile, das zweite die Spalte.
- 00 steht für den Nullzeiger und ist keine gültige Adresse.
- Der Speicher kann verwaiste Daten enthalten.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	X															
1																

Aufgabe 3

Das obere Speicherabbild enthält den Kopf (*head*) einer einfach verketteten Liste an der Adresse 0C.

Ein (Listen)Knoten besteht aus zwei benachbarten Speicherzellen. Die erste Zelle enthält einen hexadezimalen Datenwert und der zweite eine hexadezimale Adresse.

Aus dieser Datenstruktur soll effizient der Wert 05 gelöscht werden.

Stelle die *veränderten* Werte im Speicherabbild *nach* dem Löschen des Knotens im unteren Raster dar.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	X					0A	13						16			
1				05	19		1D	05		12	1D			11	00	

- Das erste Adress-Halbbyte bezeichnet die Zeile, das zweite die Spalte.
- 00 steht für den Nullzeiger und ist keine gültige Adresse.
- Der Speicher kann verwaiste Daten enthalten.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	X															
1																

Aufgabe 4

Das Speicherabbild unten enthält den Wurzelknoten eines Binärbaums an der Adresse 11.

Ein (Baum)Knoten besteht aus drei benachbarten Speicherzellen. Die erste Zelle enthält einen hexadezimalen Datenwert, die zweite die hexadezimale Adresse des linken Kindknotens und die dritte Zelle die hexadezimale Adresse des rechten Kindknotens.

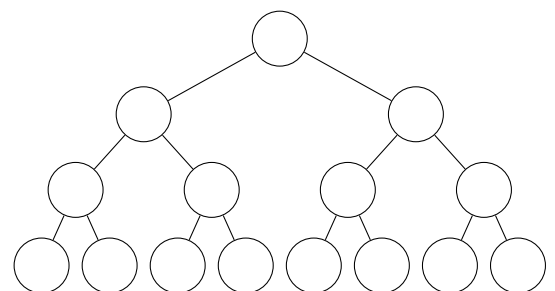
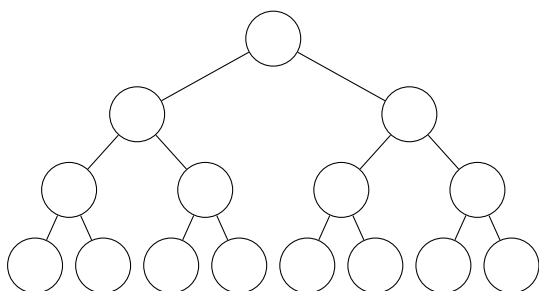
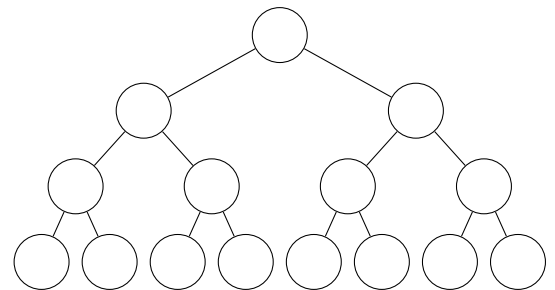
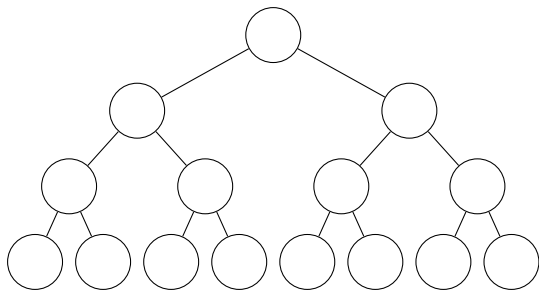
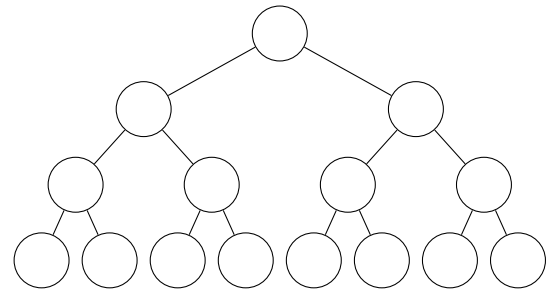
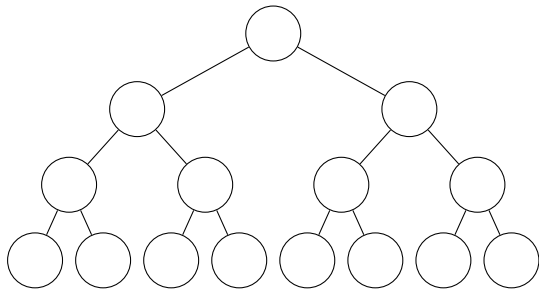
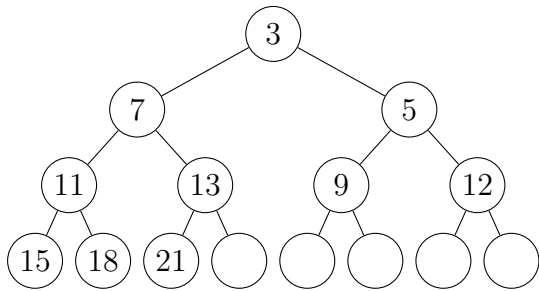
Stelle den zugehörigen Baum in graphischer Form dar.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	X	4A	21	3D	17	27	00	2F	2C	37	22	00	00	35	00	3E
1	00	1F	05	38	0C	00	10	23	00	2C	14	02	6A	1F	3B	06
2	14	3A	09	00	00	07	A5	A1	00	00	30	3E	29	00	19	35
3	00	00	15	28	22	15	26	B9	03	27	0A	0D	0F	2D	1F	23

- Das erste Adress-Halbbyte bezeichnet die Zeile, das zweite die Spalte.
- 00 steht für den Nullzeiger und ist keine gültige Adresse.
- Der Speicher kann verwaiste Daten enthalten.

Aufgabe 5

Füge den Wert 4 in den Min-Heap ein und zeige schrittweise, wie die Min-Heap-Eigenschaft wieder hergestellt wird. Verwende für jeden Schritt einen separaten Baum. Überzählige Knoten können ignoriert werden.



Aufgabe 6

Das kleinste Element wurde aus dem Min-Heap entfernt. Zeige schrittweise, wie die Heap-Struktur und die Min-Heap-Eigenschaft wieder hergestellt werden. Verwende für jeden Schritt einen separaten Baum. Überzählige Knoten können ignoriert werden.

