CTIS264 Midterm Solution 07/11/13

Name: Section: Student No:

Closed Book, closed note exam. Show your work! we must follow your reasoning. Give the best result that you can give! Over 100 points is bonus.

SIGNATURE	Time of Submission:

1. Suppose you have an algorithm $\mathcal{A}$ Input the this problem naturally divided into 4 groups, and You happen to know performance of the algorithm for each groups of input, in terms of running time, of these algorithms over all possible inputs of size n.

Group	complexity	Simplified
Group 1	$.5n^5 + 10n^4 + 100\log n + 10^9$	
Group 2	$100n^2\log n + 0.01 * n^2 + n\log n$	
Group 3	$10^6 n\log n + 10^9 + 0.01n^4 + 0.01 * 2^n$	
Group 4	$+\,1000n^2 + 10^6 + 0.01n\log n$	

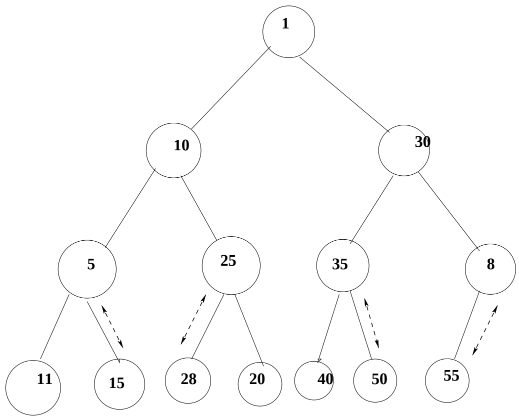
Give performance of $\mathcal{A}$ in terms of $O(.), \Omega(.), \Theta(.)$ notation. **5** pts

Solution:

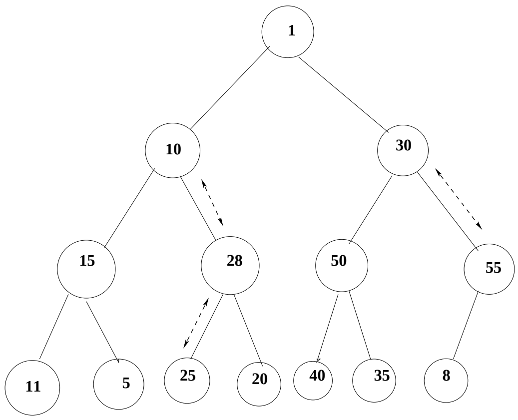
Group	complexity	Simplified
Group 1	$.5n^5 + 10n^4 + 100\log n + 10^9$	n^5
Group 2	$100n^2\log n + 0.01 * n^2 + n\log n$	$n^2\log n$
Group 3	$10^6 n\log n + 10^9 + 0.01n^4 + 0.01 * 2^n$	2^n
Group 4	$+\,1000n^2 + 10^6 + 0.01n\log n$	$\mathbf{n^2}$

THus $O(2^n)$, $\Omega(n^2)$ and Θ is not defined.

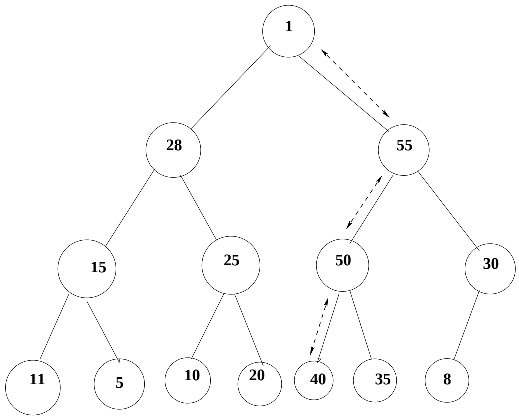
2. Build a heap containing the following items: 1 10 30 5 25 35 8 11 15 28 20 40 50 55 . **Use Heapify!** **10 pts**



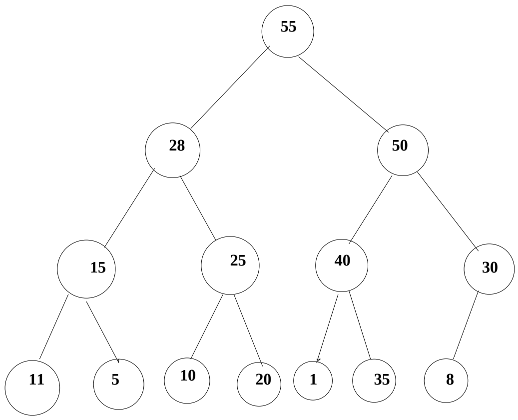
initial and level 2 swaps



level 1 swaps



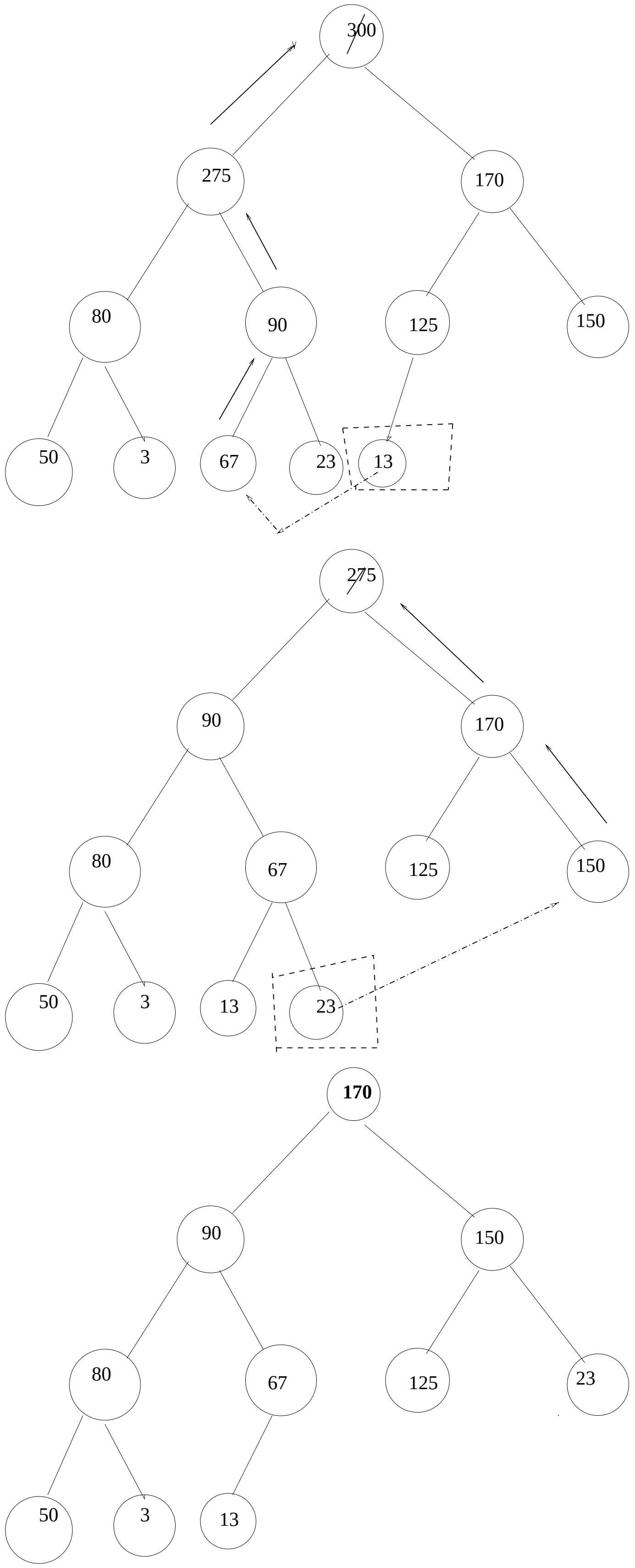
level 2 swaps



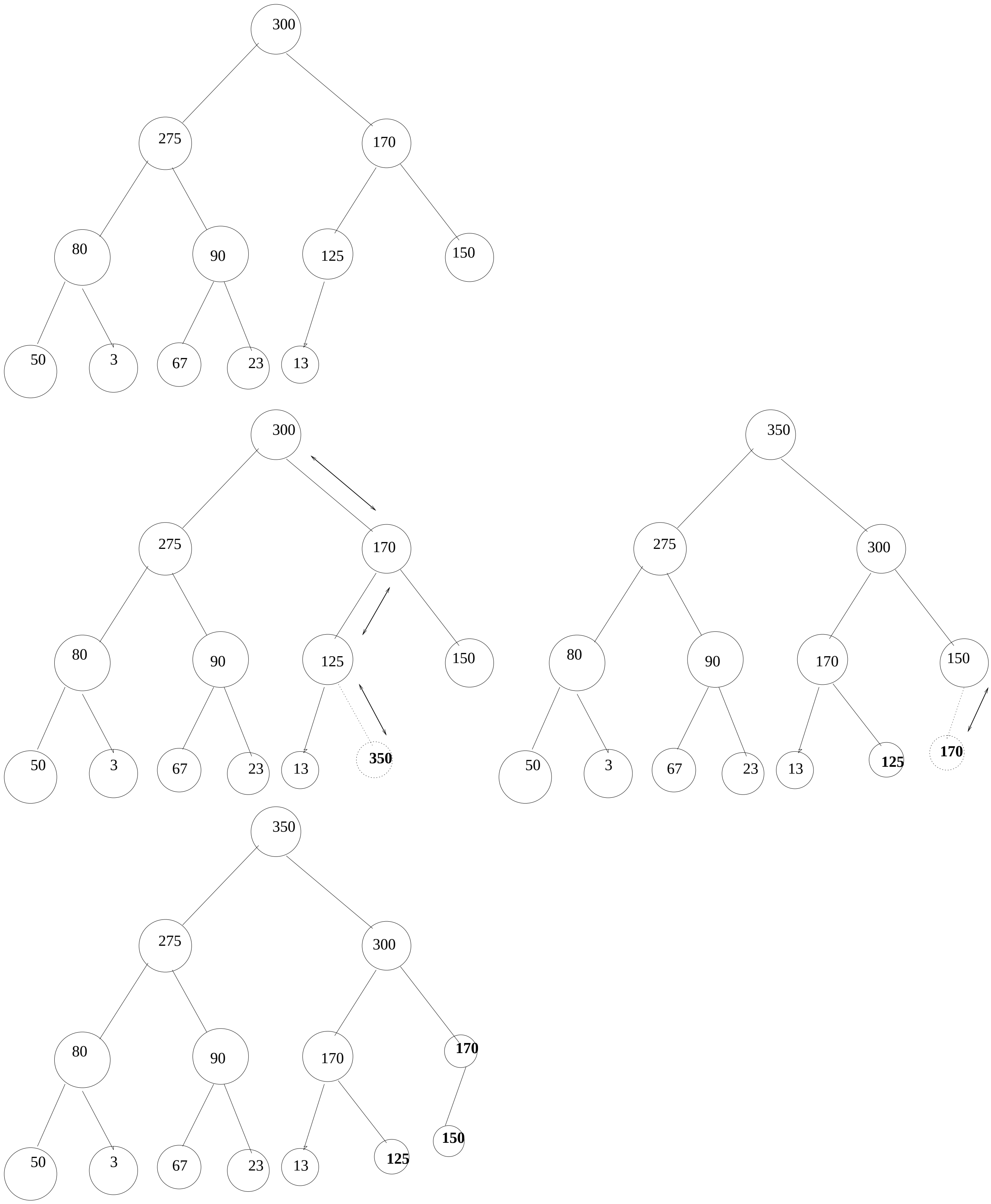
final heap

3. Given Heap H_o, apply succesively

   i) apply twice deletemax operation to H_o (**3 pts** each)



ii) insert 350, insert 170 (**2 pts** each)



4. **Hashing**. Let $h(k) = k \bmod 17$ and $h_2'(k) = k \textbf{ mod } 11$ and $h_2(k) = h_2'(k)$ if $h_2'(k) > 0$, and $h_2(k) = 1$ otherwise. Let $h(k,i) = (h_1(k) + ih_2(k))\textbf{mod}17$. Using the sequence h(k,i) i=0, 1, 2, ... We want to place following in a Hash table: 40, 21,4, 67, 31, 38, 37, 54, 6, 23, 7, 33, 50

10 pts

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Notation: h(k,i)=$h_1(k) + ih_2(k)$=t $\mapsto$ A(t)=k

First, let us compute h(k) for each k:

k	40	21	4	67	31	38	37	54	6	23	7	33	50
h_1(k)	6	4	4	16	14	4	3	3	6	6	7	16	16
h_2(k)	7	10	4	1	9	5	4	10	6	1	7	1	6
t	6	4	8	16	14	9	3	13	12	7	11	0	5

k=40 $h_1(k) = 6$, $\mapsto$ A(6)=40
k=21 $h_1(k) = 4$, $\mapsto$ A(4)=21
k=4 $h_1(k) = 4 = h_2(k)$, 4 + 4 = t , i=1, $\mapsto$ A(8)=4
k=67 $h_1(k) = 16$, $\mapsto$ A(16)=67
k=31 $h_1(k) = 14$, $\mapsto$ A(14)=31
k=38 $h_1(k) = 4$, $h_2(k) = 5$, 4+5=t, i=1, $\mapsto$ A(9)=38
k=37 $h_1(k) = 3$, $\mapsto$ A(3)=37
k=54 $h_1(k) = 3$, $h_2(k) = 10$, 3+10=t, i=1, $\mapsto$ A(13)=54
k=6 $h_1(k) = 6$, $h_2(k) = 6$, 6+6=t, i=1, $\mapsto$ A(12)=6
k=23 $h_1(k) = 6$, $h_2(k) = 1$, 6+1=t, i=1, $\mapsto$ A(7)=23
k=7 $h_1(k) = 7$, $h_2(k) = 7$, 7, 14, 21=4, 11=t, i=3, $\mapsto$ A(11)=7
k=33 $h_1(k) = 16$, $h_2(k) = 1$, 16+1=0=t, i=1, $\mapsto$ A(0)=33
k=50 $h_1(k) = 16$, $h_2(k) = 6$, 16+6=5=t, i=1, $\mapsto$ A(5)=50

Thus we obtain the table:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
33			37	21	50	40	23	4	38		7	6	54	31		67

– Solve the same input with h_1 and linear probing

5 pts

0	1	2	3	4	5	6	7	8	9	1 0	1 1	1 2	1 3	1 4	1 5	1 6

First, let us compute h(k) for each k:

k	40	21	4	67	31	38	37	54	6	23	7	33	50
h(k)	6	4	4	16	14	4	3	3	6	6	7	16	16

For each k if h(k) is empty, you plave k at postion h(k), otherwise you increase h(k) by 1 in a circular fashion, until you find an empty spot.

In terms of double hashing $h_2(k) = 1$ for each k.

Thus we obtain the table:

0	1	2	3	4	5	6	7	8	9	1 0	1 1	1 2	1 3	1 4	1 5	1 6
33	50		37	21	4	40	38	54	6	23	7			31		67

— Solve the same input with $h(x) = x \bmod 11$ and chaining **5 pts**

Let us compute $h(x)$ for each x:

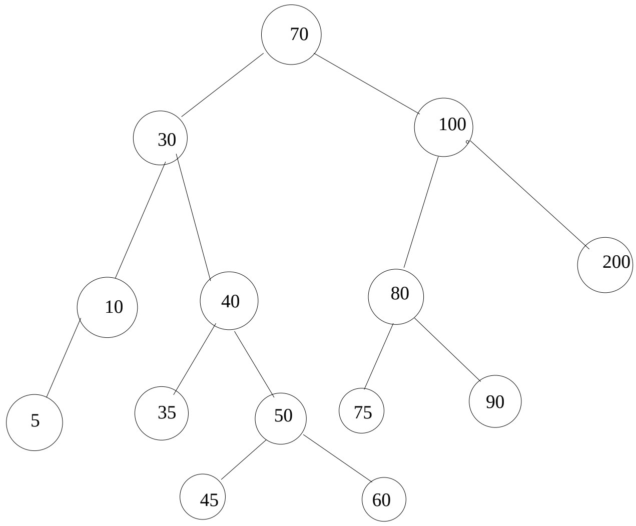
x	40	21	4	67	31	38	37	54	6	23	7	33	50
h(x)	7	10	4	1	9	5	4	10	6	1	7	0	6

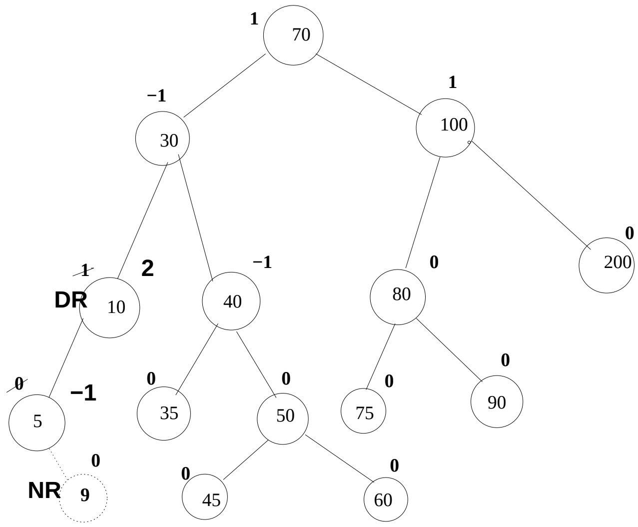
We form a linked list for each possible h(k)

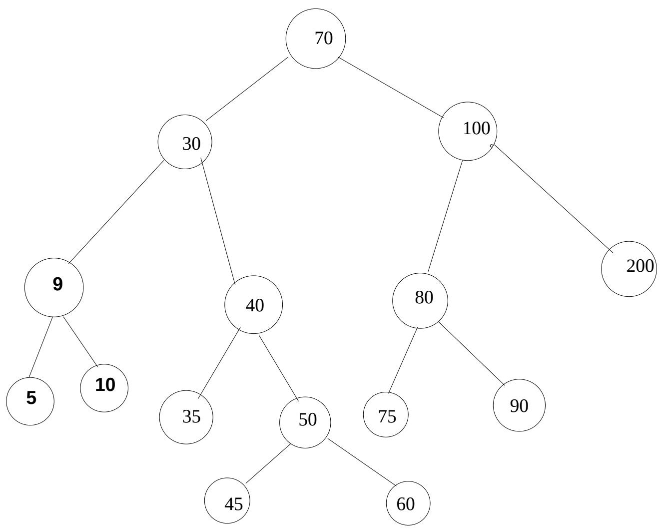
0	1	2	3	4	5	6	7	8	9	10
↓	↓			↓	↓	↓	↓		↓	↓
33	**67**			**4**	**38**	**6**	**40**		**31**	**21**
	↓			↓		↓	↓			↓
	23			**37**		**50**	**7**			**54**

5. **Search Tree** Given the AVL Search tree T_o **5 pts each**

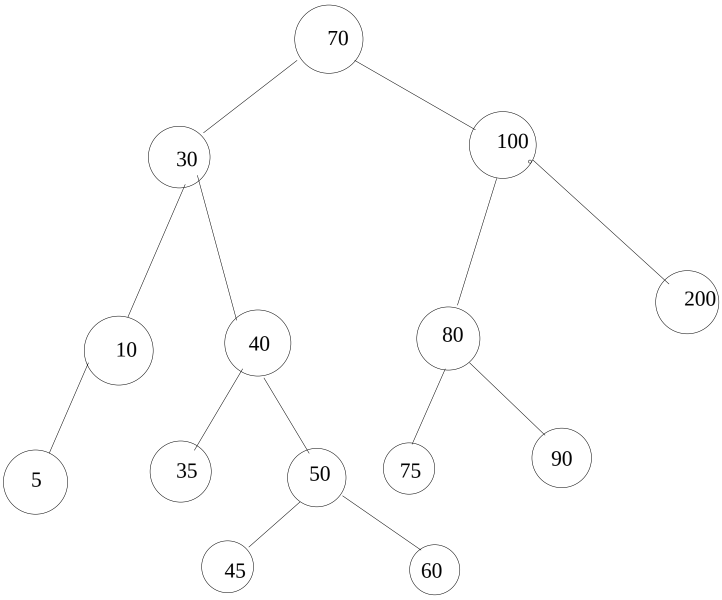
- Insert 9 into T_o Check balances after insertion. Rotate if necessary

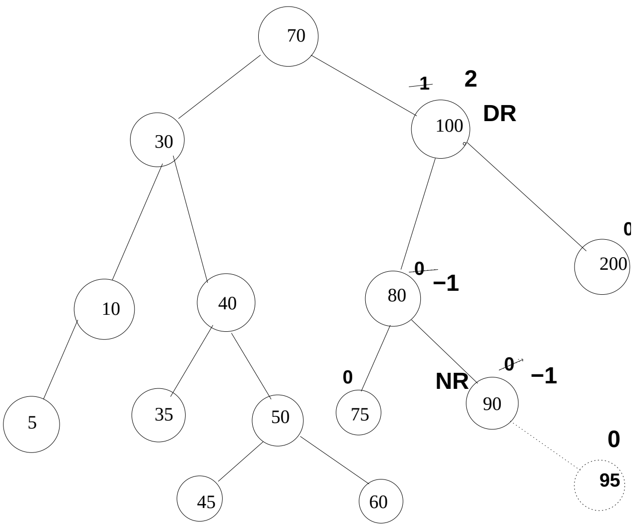


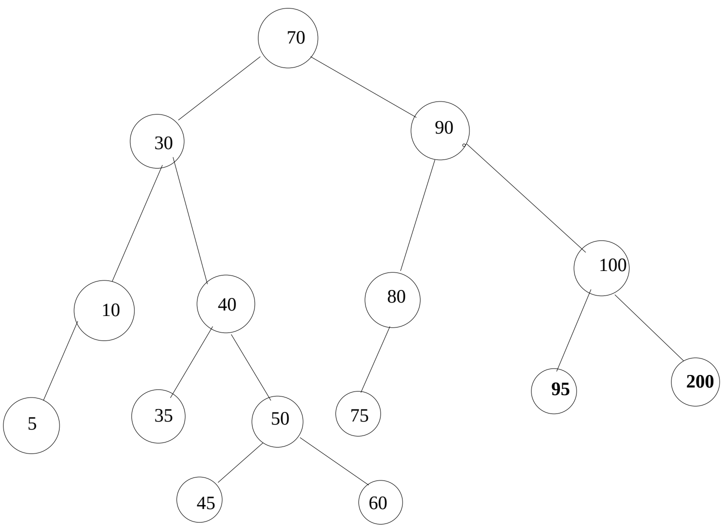




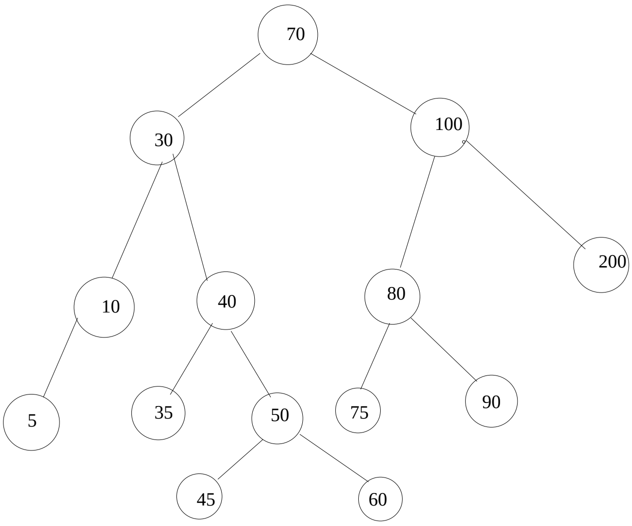
- Insert 95 into T_o Check balances after insertion. Rotate if necessary

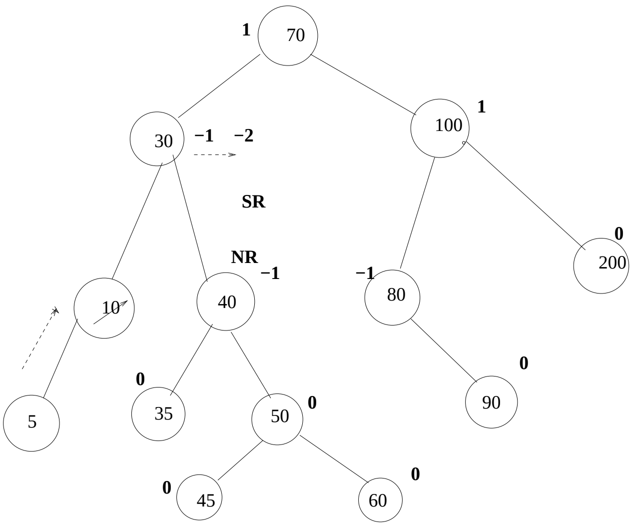


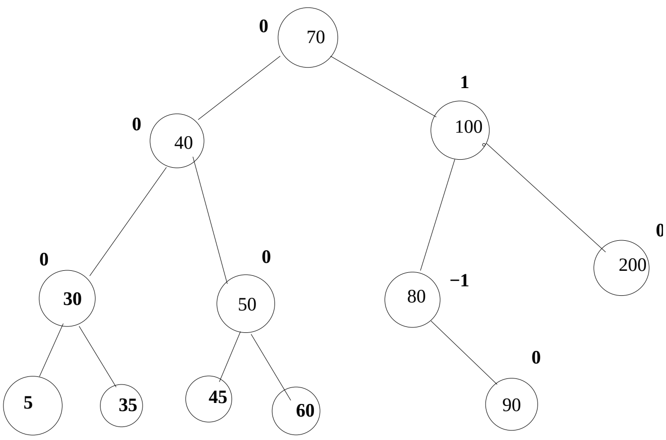




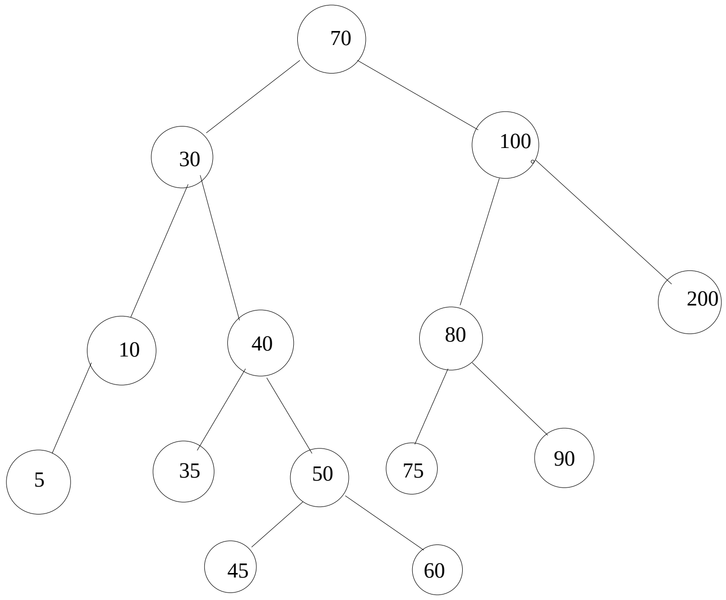
- Given the AVL Search tree T_o delete 10; rotate if necessary.

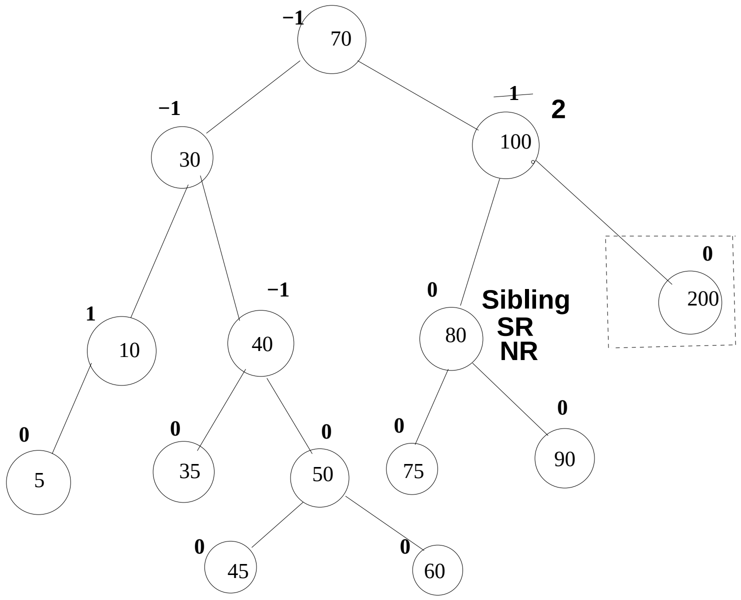


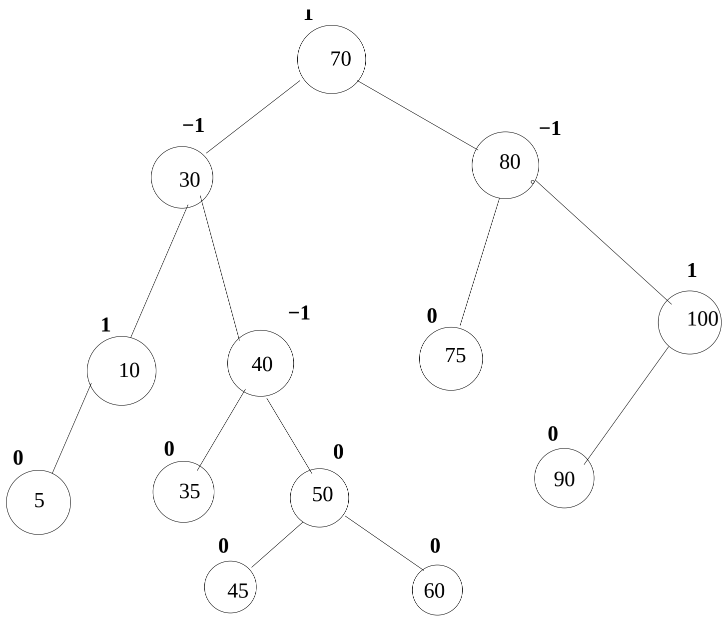




- Given the AVL Search tree T_o Starting from T_o again, delete 200; rotate if necessary.







6. Assume X is an algebraic object whose multiplication is relatively expensive. We want to compute X^n for various n.

- **A**
  Assume you can store as many X^j as you wish, how would you compute X^n for n=101, 300, 544. Indicate which X^j you store and how do you compute desired X^n's ? (Show which multiplications) **6 pts**
- **B** Given b(0), b(1), b(k), binary expansion of n, with b(k)=1, how would you compute X^n without storing all intermediate results. Write an algorithm/pseudo-code ! **4 pts**

  Solution:

  Solution:

- **A**
  n=101=64+32+4+1=$2^6+2^5+2^2+1$, $X^{97} = X^{64}*X^{32}*X^4*X$. Thus we need 6+3=9 multiplication
  n=300=256+32+8+4=$2^8+2^5+2^3+2^2$, and $X^{300} = X^{256}*X^{32}*X^8*X^4$. Thus we need 8 + 3 =11 multiplication
  n=544=512+32=2^9+2^5, and $X^{536} = X^{512}*X^{32}$. Thus we need 9+1=10 multiplication
- **B**

```
T=X; P=1
for t=0,...k
if b(t)=1 then P <-- P * T
T <-- T * T
endfor
```

P return the required product with one extra multiplication. We can eliminate extra multiplication via these alternatives.

```
T=X; P=1
if b(0)=1  P <-- X
for t=1,...k
T <-- T * T
if b(t)=1 then P <-- P * T
endfor
```

```
T=X; P=1
for t=0,...k-1
if b(t)=1 then P <-- P * T
T <-- T * T
endfor
P <--  P * T
```

Compute number of multiplications in terms of b(i)'s

k+b(0)+b(1)+ . . . + b(k-1)

7. **Median** Given the list, we want to find 10'th element of the list (in the sorted list).

Take the first element of the list in question as the pivot element, and use the partition algorithm as discussed in class. For each recursive call to partition algorithm, indicate pivot element, (with a circle) the list input to partition algorithm, and order of the element which is sought (e.g. 5th, 7th etc). Where n denotes size of the list, k order of the element sought, p is the pivot element

10 pts

n	k	p	List	partition
12	10		13 19 4 12 25 5 8 3 30 17 9 20	
12	10	13	13 19 4 12 25 5 8 3 30 17 9 20	(9 4 12 5 8 3) 13 (19 30 17 25 20)
5	3	19	19 30 17 25 20	(17) 19 (30 25 20)
3	1	30	30 25 20	(25 20) 30 ( )
2	1	25	25 20	(20) 25 ()
1	1		**20**	

[let $\beta = \log_b a$ and α such that $f(n) = n^\alpha \log^k n$ in the formulation $T(n) = aT(n/b) + f(n)$)

- $$T(n) = 2T(n/3) + \Theta(n^2)$$

  $\beta \log_3 2 < 1 < 2 = \alpha$ Thus T(n)=f(n)=$O(n^2)$

- $$T(n) = 6T(n/2) + \Theta(n^2)$$

  $\beta = \log_n 6 > 2 = \alpha$. Then T(n)=$n^\beta = n^{\log_2^6}$

- $$T(n) = T(n/4) + \Theta(\log^2 n)$$

  $\beta = \log_4 1 = 0 = \alpha$ and $k = 2 \geq 0$. Thus T(n)=$f(n)\log n = \log^3 n$

- $$T(n) = 2T(n/3) + \Theta(n \log n]$$

  $\beta = \log_3 2 < 1 = \beta$ and $k = 1 > 0$. Master Theorem Does not apply !
  (It applies if $\alpha = \beta, k \geq 0$ OR $\alpha \neq \beta$ and k=0)

- $$T(n) = 3T(n/3) + \Theta(n/ \log n]$$

  Since $k = -1 < 0$ Master Theorem Does not apply ! (even though $\alpha = \beta$) $\beta = \log_3 3 = 1 = \alpha$

- $$T(n) = 3T(n/3) + \Theta(n \log n]$$

  $\beta = \log_3 3 = 1 = \alpha$ and $k = 1 > 0$. Thus $T(n) = O(f(n)\log n) = O(n(\log n)^2)$

- $$T(n) = 3T(n/3) + \Theta(\sqrt{n}]$$

  $\beta = \log_3 3 = 1 > \alpha = 1/2$ and $k = 0$. Thus
  $T(n) = O(n^\beta) = O(n)$