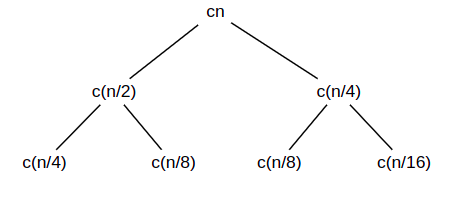
**Problem 1.**



the height of the tree is log2(n)

T(n) = O(n) was the solution for this recurrence

**Problem 2.**

1. T (n) = 3T ( n/3 ) + n/2

According to Master Theorem, a=3, b=3, f(n)=n/2

n^(log3(3)) = n

=> case 2

T (n) = Θ(nlg(n))

2. T (n) = 2T ( n/4 ) + n0.51

According to Master Theorem, a=2, b=4, f(n)=n0.51

n^(log4(2) + *ε*) = n0.51

*ε*=0.01

=> case 3

if: 2f (n/4) ≤ cf (n) => 2/c ≤ 40.51 => c ≥ 0.986

Choose c=0.99 < 1

T (n) = Θ(n0.51)

3. T (n) = 4T ( n/2 ) + n2

According to Master Theorem, a=4, b=2, f(n)=n2

n^(log2(4)) = n^2

=> case 2

T (n) = Θ(n2lg(n))

4. T (n) = 4T ( n/2 ) + n

According to Master Theorem, a=4, b=2, f(n)=n2

n^(log2(4) - *ε*) = n

=> case 1

*ε*=1

=> T (n) = Θ(n2)

5. T (n) = 7T ( n/3 ) + n2

According to Master Theorem, a=7, b=3, f(n)=n2

n^(log3(7) + *ε*) = n2

*ε*=0.23

=> case 3

if: 7f (n/3) ≤ cf (n) => c ≥ 7/9

Choose c=0.99 < 1

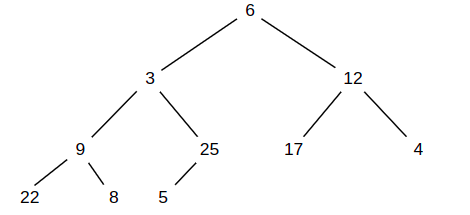
T (n) = Θ(n2)

**Problem 3.**

1.A = [6,3,12,9,25,17,4,22,8,5]

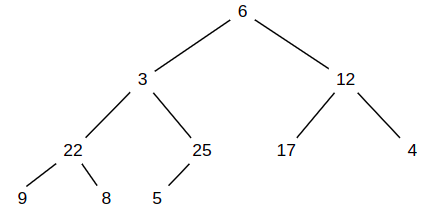
To build the heap, heapify only the nodes:

[6, 3, 12, 9, 25] in reverse order.

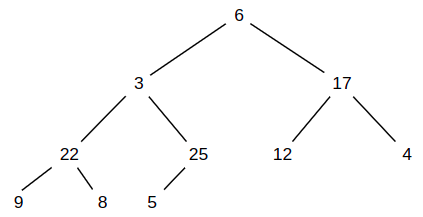


Heapify 25: Not change

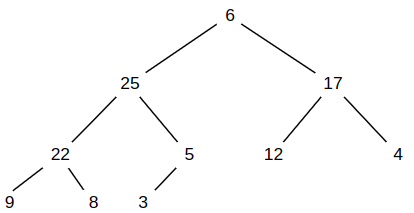
Heapify 9: Swap 22 and 9



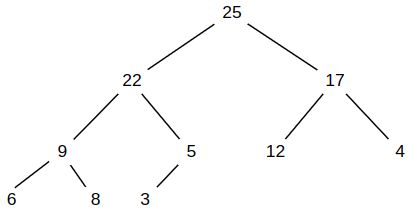
Heapify 12: Swap 17 and 12



Heapify 3: Swap 25 and 3, then swap 3 and 5



Heapify 6: Swap 25 and 6, then swap 6 and 22, then swap 9 and 6

****

2.

We have the MAX-HEAP algorithm like this:

*for i=heapsize/2 to 1:*

*while node < leftnode or node < rightnode:*

*swap*

*move up*

*endwhile*

*endfor*

We can see the while loop cost O(lg(n)) and the for loop cost O(n)

So the complexity time is O(nlg(n))

**Problem 4.**

Assume n-element heap has height *h*

*2h+1 - C = n* with C is a constant

C = n – 2floor(log(n))

=> *2h+1 +* 2floor(log(n)) = 2n

=> h = log(2n - 2floor(log(n)) ) - 1

**Problem 5.**

Since T (n) = T (n − 1) + Θ(n)

So there is a constant c > 0 such that

T(n) ≥ T(n-1) + cn

For all large n, we guess T(n) = O(n2)

=> T(n) ≤ kn2 with k > 0

=> T(n) ≤ k(n-1)2 + cn ≤ kn2

=> kn2 - 2kn + k + cn ≤ kn2

=> k ≥ cn/(2n-1)

max of n/(2n-1) is 1 with n ≥ 1

Therefore, any k ≥ c is satisfy

So, with n ≥ 1, T(n) = Θ(n2)