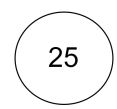
CS 601 - Advanced Algorithms
Assignment 3

Question 1.
Consider the array below:
25, 17, 36, 2, 3, 100, 1, 19, 7.

a) Construct a binary search tree (BST) out of the given array by using insert operation as discussed and practiced in the class. You are going to insert 25 first, then you insert 17 to the tree, so on. Show all your work step by step. In each step (i) draw the corresponding tree as well as (ii) the resulting array.

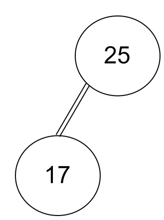
Iteration 1 ⇒ Inserting 25

25								



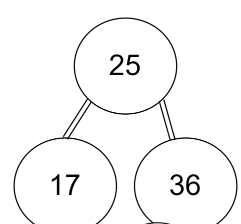
Iteration 2 ⇒ Inserting 17

25	**17**							



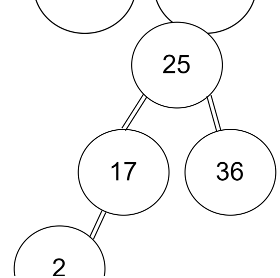
Iteration 3 ⇒ Inserting 36

25	17	**36**						



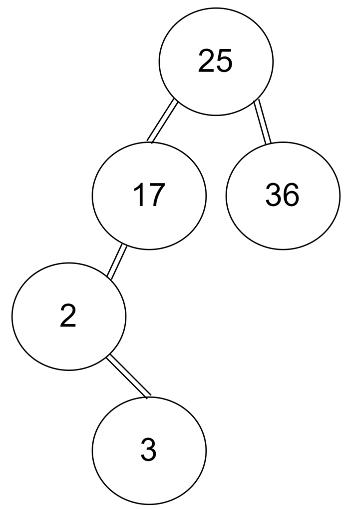
Iteration 4 ⇒ Inserting 2

25	17	36	**2**					



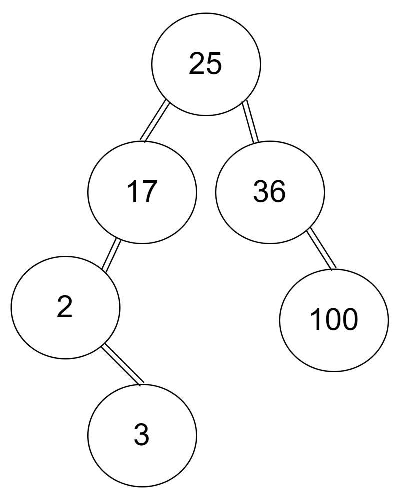
Iteration 5 ⇒ Inserting 3

25	17	36	2					**3**

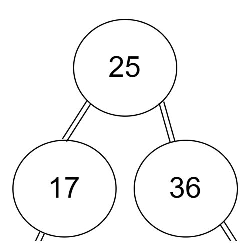


Iteration 6 ⇒ Inserting 100

25	17	36	2			**100**		3



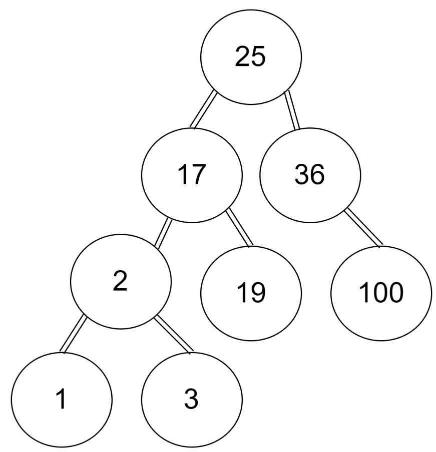
Iteration 7 ⇒ Inserting 1



25	17	36	2			100	**1**	3

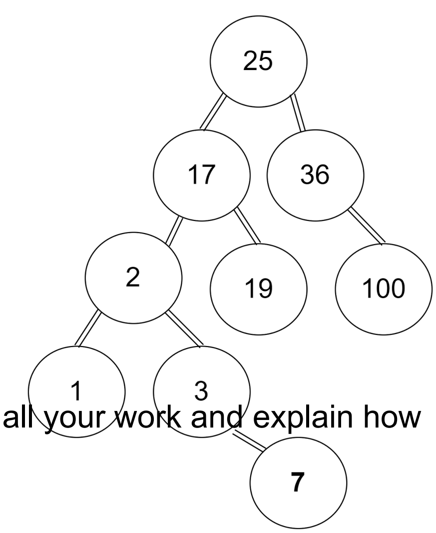
Iteration 8 ⇒ Inserting 19

25	17	36	2	**19**		100	1	3

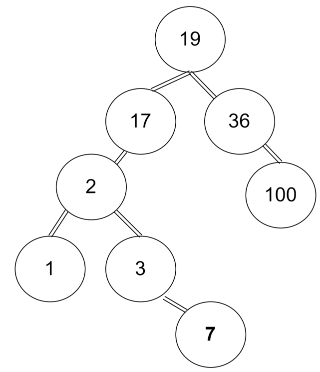


Iteration 9 ⇒ Inserting 7

0								8	(9	-17)	18
25	17	36	2	19		100	1	3			**7**



b) Delete 25 from the BST constructed in part (a). Show all your work and explain how that works.



Deleting 25 from the constructed binary search.
25 is the root element. If we removed the root element,
Either the maximum value of the left subtree is replaced to the position of deleted element or the minimum value of the right subtree is replaced to the position of the deleted element.

0								8	9-17		18
19	17	36	2			100	1	3			**7**

⇒ Replaced the maximum value (19) from the left subtree to the position of the deleted element (25).

Question 2
Let f(n)= 300 n^2 log n+ 2n^6 +2000. Replace a correct function instead of each question mark
below. Justify clearly.
a) f(n)=O(?)
b) f(n)=Ω(?)
c) f(n)=θ(?)

Given f(n)= 300n^2 logn + 2n^6 + 2000

a) f(n)=O(?)

f(n) = O(g(n)) if there is a positive constant C and n0

So f(n)<=c*g(n) for every n>=n0

We can write 300n^2 log n + 2n^6 + 2000 <= 300n^6 + 2n^6 + 2000 n^6

300n^2 log n + 2n^6 + 2000 <= 2302 n^6

Where c=2302 and n0=1

This shows f(n) <= c*g(n) for n>=1
According to f(n)=O(g(n))

$f(n) = O(n^6)$

Here n^6 is closest value, big O notation can be any value greater than n^6

b) $f(n)=\Omega(?)$

$f(n) = \Omega(g(n))$ if there is a positive constant C and n_0

So $f(n) \geq c \cdot g(n)$ for every $n \geq n_0$

We can write $300 n^2 \log n + 2n^6 + 2000 \geq n^6$

Where $c=1$ and $n_0=1$

This shows $f(n) \geq c \cdot g(n)$ for $n \geq 1$

According to $f(n)=\Omega(g(n))$

$f(n) = \Omega(n^6)$

Here n^6 is closest value, omega notation can be any value lesser than n^6

c) $f(n)=\theta(?)$

$f(n) = \theta(g(n))$ if there is a positive constant c_1, c_2 and n_0

So $c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$ for every $n \geq n_0$

We can write $n^6 \leq 300n^2 \log n + 2n^6 + 2000 \leq 300n^6 + 2n^6 + 2000 n^6$
Where $c_1=1$, $c_2=2302$ and $n_0=1$

This shows $c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$ for $n \geq 1$
According to $f(n)=\theta(g(n))$

$f(n) = \theta(n^6)$

For $\theta(n)$ there will not be the closest value, because it has an upper and lower bound the value is exact.

Question 3

. Consider an instance of the stable matching problem with 4 men and 4 women. Provide their preference lists as you would like (show them using two tables as in the slides).Then provide an assignment that is *unstable*. Now provide an assignment that is *stable*. For each case, explain in detail why the assignment is stable/unstable.

4 men and their preference list:

	1st	2nd	3rd	4th
Adam(A)	E	F	G	H
Bob(B)	H	F	G	E
Charles(C)	G	E	H	F
David(D)	F	E	H	G

4 women and their preference list:

	1st	2nd	3rd	4th
Emily(E)	A	B	C	D
Flora(F)	B	A	D	C
Gracie(G)	C	B	A	D
Hope(H)	B	D	C	A

⇒ **Unstable Assignment is {(A,E),(B,G),(C,F),(D,H)}**
An assignment is not stable when there is at least one unstable pair.
From the mentioned assignment

⇒ **A - E stable pair.**

For B-G and C-F, we can find other better pairs B-F can hook up and C-G can hook up because they have better preference than B-G and C-F. This shows both the pairs are not stable.

For D-H, we can find a better pair, B-H can hook up than D-H is also not a stable pair.

Hence overall assignment is not stable.

⇒**Stable Assignment is {(A,E),(B,H),(C,G),(D,F)}**

For A-E, A prefers E most and similarly E prefers A most.

For B-H, B prefers H most and similarly H prefers B most.

For C-G, C prefers G most and similarly G prefers C most.

For D-F, D prefers F most
But F has B in 1st Preference but B has H 1st preference
And also F has A in 2nd Preference but A has E has 1st preference
Now F has D in the next preference so it is a stable assignment.

Question 4:

Let hospitals be h and students be s

We can come up with algorithm similar to GS algorithm for Stable matching

Algorithm is as follows

Initialize s and h to free
while (hospital has available positions) {
  h = current hospital
  s = 1st Student on h's list for which h has not visited yet
  if(s is free)
    assign h to s and decrement available position in hospital h by 1
  else if (s prefers h than previous assignment h')
    s will be assigned to h and available position in h' is increased
  else
    s will reject h
}

Question 5 : Let f(n)= 3 log n5+ n2 +14, then f(n) is O(?). Justify.

Given f(n)= 3 log n^5 + n^2 +14

⇒ **f(n)=O(?)**

$f(n) = O(g(n))$ if there is a positive constant C and $n0$

So $f(n) <= c*g(n)$ for every $n >= n0$

We can write $3 \log n^5 + n^2 + 14 <= 3 n^2 + n^2 + 14 n^2$ [$\log n^5 < n^2$]

$3 \log n^5 + n^2 + 14 <= 18 n^2$

Where $c=18$ and $n0=1$

This shows $f(n) <= c*g(n)$ for $n>=1$

According to $f(n)=O(g(n))$

$f(n) = O(n^2)$

Here n^2 is closest value, big O notation can be any value greater than n^2