## Question 1

a) Convert binary number 1011 0101 into Hex. Show your workings:

1011 = upper nibble
0101 = lower nibble
$1011 = 1\times2^3 + 0\times2^2 + 1\times2^1 + 1\times2^0 = 8+0+2+1 = 11 = B$ (hex)
$0101 = 1\times2^2 + 1\times2^0 = 5$ (hex)
1011 0101 (bin) = B5 (hex)

b) Convert binary number "1001 0111" into decimal. Show your working.

Do we assume leftmost digit represents a neg.?

$1001 = 1\times2^3 + 1\times2^0 = 8+1 = 9$ (hex)
$0111 = 1\times2^2 + 1\times2^1 + 1\times2^0 = 4+2+1 = 7$ (hex)
$9\times16^1 + 7\times16^0 = 144+7 = 151$ (dec)

516
x 9
144

c) Convert "D4" Hex into decimal. Show your working.

D = 13 (hex)
4 = 4 (hex)
$13\times16^1 + 4\times16^0 = 208 + 4 = 212$ (dec)

16
x13
48
+160
208

d) Specify for each of the following whether they are stored in the Stack or the Heap?

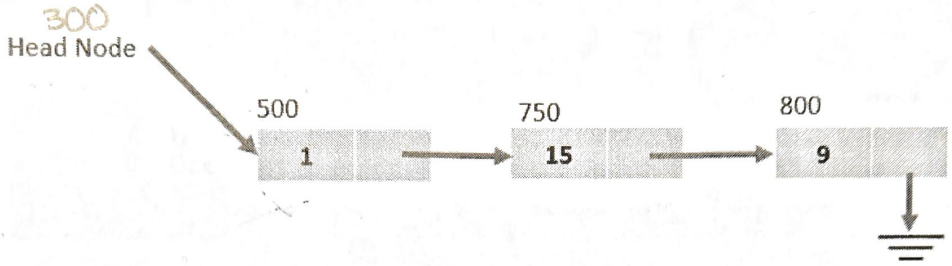
lowercase i in "integer" = primitive not wrapper class

| Data type | Stack or Heap? |
|---|---|
| integer | Stack |
| String | Heap |
| Array | Heap |

**Question 2**

For the Linked List shown below, complete the Address and Contents columns of the memory table for the head node pointer and all the nodes. (You may optionally use the Comments column if you need to). Assume that the head node pointer is stored in location 300, and values stored in each node are long integers that take up 8 bytes and each pointer address also takes up 8 bytes of space. The starting address of each node is indicated. NULL values may simply be indicated as 'NULL'.

head pointer @300 is of type node




Do comments have to look like how they do in the ans key even though they are optional?

| Address (in decimal) | Contents | (Comment) |
|---|---|---|
| 300 - 307 | @500 | contents are stored at address 500 — head node pointer |
| 500 - 507 | 1 | 1st node |
| 508 - 515 | @750 | contents are stored at address 750 — 1st node |
| 750 - 757 | 15 | 2nd node |
| 758 - 765 | @800 | contents are stored at address 800 — 2nd node |
| 800 - 807 | 9 | 3rd node |
| 808 - 815 | null | contents consist of a null address — 3rd node |
| | | |
| | | |
| | | |

head pointer also has to be stored @ an address of the specified length

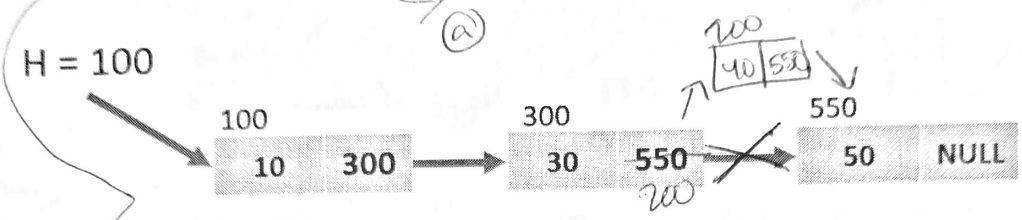
**Question 3**

Consider a sorted List of integers: List=[10, 30, 50], which is stored as a sorted Linked List data structure as shown below. Show how the linked list would change after:

a) We insert a new node with content "40" while ensuring that the list continues to be sorted. Assume that this new node is stored starting at memory address 200.

b) What would be the best case, worst case, and average time-complexity in Big-O notation for searching for a key in such a list with N elements?

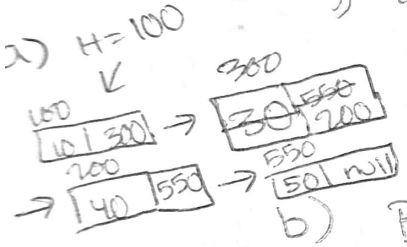
sorted linked list, insertion




Question asked to show

a) The pointer field of the node at memory address 300 would contain the memory address 200 instead of 550. The new node at address 200 would have a pointer field containing the address to the node at memory address 550.

contents, in this case node, at




b) Best case:

Worst case: O(n)

Average case: O(n)

* review complexities of insertion & deletion of a sorted linked list => review the table *

Best case: O(1)

Worst case: O(n)

Average case: O(n)

Scratch

$O(n)+O(1) = O(n+1) = O(n)$

$O(\frac{n}{2})+O(1) = O(\frac{n+2}{2})$

$= O(\frac{1}{2}(n+2))$

$= O(\frac{1}{2}n+1)$

$= O(\frac{1}{2}n)$

$= O(n)$

Scratch

$O(\frac{n}{2}) =$

$O(\frac{1}{2}n)$

$= O(n)$

essentially doing a linear search