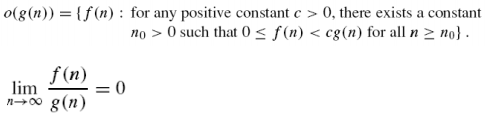
Q1(a) Explain the meaning of 'Little o' Notation? [2 Marks]

Ans:



*o*-notition is used to denote an upper bound that is not tight.

2*n* = *o*(*n*2), but 2*n*2 ≠ *o*(*n*2).

(b) Find where the indicated elements of an array A are stored, if the base address of A is 200 and LB=0.

a) double A[6]

b) int A[28]Assume that int(s) are stored in 4 bytes and doubles(s) in 8 bytes. [2 Marks]

Ans:

Loc (X[i]) = Loc(X[LB]) + w\*(i-LB)

1. Loc (A[6]) = 200 + 4\*(6-0)

=224

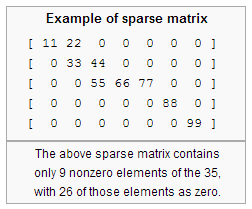
1. Loc (A[28]) = 200 + 8\*(28-0)

= 424

(c) What is Sparse Matrix . Explain with example. [2 Marks]

Ans:

In the subfield of [numerical analysis](http://en.wikipedia.org/wiki/Numerical_analysis), a sparse matrix is a [matrix](http://en.wikipedia.org/wiki/Matrix_(mathematics)) populated primarily with zeros as elements of the table.



(d) What do you mean by garbage collection? [2 Marks]

Ans:

In [computer science](http://en.wikipedia.org/wiki/Computer_science), garbage collection (GC) is a form of automatic [memory management](http://en.wikipedia.org/wiki/Memory_management). The garbage collector, or just collector, attempts to reclaim [garbage](http://en.wikipedia.org/wiki/Garbage_(computer_science)), or memory occupied by[objects](http://en.wikipedia.org/wiki/Object_(computer_science)) that are no longer in use by the [program](http://en.wikipedia.org/wiki/Application_software).

(e) Write recursive procedure to calculate nth term of Fibonacci sequence. [2 Marks]

Ans:

In the Fibonacci sequence, the basis cases are n = 0 and

n = 1. Since the sequence is only deﬁned for nonnegative

integers n, the recursive deﬁnition will always approach 0.

Basis cases:

F0= 1

F1= 1

Recursive case (n>=2):

Fn = Fn-1 + Fn-2

(f) Differentiate Adjacency Matrix and Path Matrix. [2 Marks]

Ans:

Let G = (V,E) be a graph with n vertices. The adjacency matrix of G is a two-dimensional   
n by n array, say adj-mat. If the edge (vi, vj) is in E(G), adj-mat[i][j]=1. If there is no such edge in E(G), adj-mat[i][j]=0.

If Adjacency Matrix is A and B is path matrix then,

(g) Write an algorithm for evaluation of a postfix expression using stacks. [2 Marks]

Ans:

Algorithm:

1. Traverse the left subtree of R in postorder.

2. Traverse the right subtree of R in postorder.

3. Process the root R.

or

POSTORD(INFO, LEFT, RIGHT, ROOT)

1. Set TOP:= 1, STACK[1]:= NULL and PTR := ROOT.
2. Repeat Step 3 to 5 while PTR!=NULL:
3. Set TOP := TOP+1, STACK[Top]:=PTR.
4. If RIGHT[PTR]!=NULL, then:

Set TOP := TOP+1, Stack[Top]:=-RIGHT[PTR].

[End of If structure]

1. Set PTR:=LEFT[PTR]

[End of step 3 Loop]

1. Set PTR:=STACK[Top], TOP := TOP-1.
2. Repeat while PTR>0:

a)Apply PROCESS to INFO[PTR].

b)PTR:=STACK[Top], TOP := TOP-1.

[End of Loop]

1. Repeat while PTR<0:
   1. Set PTR:= -PTR.
   2. Go to Step 2.

[End of If structure]

1. Exit.

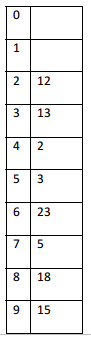
(h) What are left and right skewed BSTs? State their disadvantage. [2 Marks]

left skewed BST right skewed BST

The main disadvantage of left and right skewed BSTs is that the worst time complexity of search is O(n).

Q8. (a)The keys 12,18,13,2,3,23,5 and 15 are inserted into an initially empty hash table of length 10 using open addressing with hash function h(k)=k mod 10 and linear probing . What is the resultant hash table? [5 Mark]

Ans:



(b)What is linear probing and how it is different from quadratic probing? How can we resolve the collision (Explain with the help of an example)? [(3+7)Marks] [15 Marks]

Ans:

Linear probing is a scheme in computer programming for resolving hash collisions of values of hash functions by sequentially searching the hash table for a free location.

Given an ordinary hash function H(x), a linear probing function (H(x, i)) would be:

 H(x, i) =  (H(x) + i) \pmod n.\, 

Here *H(x)* is the starting value, *n* the size of the hash table, and i is 1, 2, 3….

This algorithm, which is used in open-addressed hash tables, provides good memory caching (if i is equal to one), through good locality of reference, but also results in clustering, an unfortunately high probability that where there has been one collision there will be more. The performance of linear probing is also more sensitive to input distribution when compared to quadratic probing.

Given an ordinary hash function H(x), a quadratic probing function (H(x, i)) would be:



Can resolve the collision by taking example of solution of part a) and it is required to explain the collision and then the process of resolution of collision.