MCQ

1. Operating Systems | Set 1

Following questions have been asked in GATE CS exam.

1. Which of the following is NOT a valid deadlock prevention scheme? (GATE CS 2000)
(a) Release all resources before requesting a new resource
(b) Number the resources uniquely and never request a lower numbered resource than the last one requested.
(c) Never request a resource after releasing any resource
(d) Request and all required resources be allocated before execution.

Answer: (c)
References:
http://www.cs.jhu.edu/~yairamir/cs418/os4/sld013.htm
http://en.wikipedia.org/wiki/Deadlock

2. Let m[0]…m[4] be mutexes (binary semaphores) and P[0] …. P[4] be processes. Suppose each process P[i] executes the following:

```
wait (m[i]); wait(m[(i+1) mode 4]);

------

release (m[i]); release (m[(i+1)mod 4]);
```

This could cause (GATE CS 2000)
(a) Thrashing
(b) Deadlock
(c) Starvation, but not deadlock
(d) None of the above

Answer: (b)
Explanation:
You can easily see a deadlock in a situation where..
P[0] has acquired m[0] and waiting for m[1]
P[1] has acquired m[1] and waiting for m[2]
P[2] has acquired m[2] and waiting for m[3]
P[3] has acquired m[3] and waiting for m[0]

3. A graphics card has on board memory of 1 MB. Which of the following modes

can the
card not support? (GATE CS 2000)
(a) 1600 x 400 resolution with 256 colours on a 17 inch monitor
(b) 1600 x 400 resolution with 16 million colours on a 14 inch monitor
(c) 800 x 400 resolution with 16 million colours on a 17 inch monitor
(d) 800 x 800 resolution with 256 colours on a 14 inch monitor

Answer: (b)
Explanation:
Monitor size doesn't matter here. So, we can easily deduct that answer should be (b) as this has the highest memory requirements. Let us verify it.
Number of bits required to store a 16M colors pixel = ceil(log2(16*1000000)) = 24
Number of bytes required for 1600 x 400 resolution with 16M colors = (1600 * 400 * 24)/8 which is 192000000 (greater than 1MB).

4 Consider a virtual memory system with FIFO page replacement policy. For an arbitrary page access pattern, increasing the number of page frames in main memory will (GATE CS 2001)
a) Always decrease the number of page faults
b) Always increase the number of page faults
c) Some times increase the number of page faults
d) Never affect the number of page faults

Answer: (c)
Explanation:
Incrementing the number of page frames doesn't always decrease the page faults (Belady's Anomaly). For details see http://en.wikipedia.org/wiki/Belady%27s_anomaly

5. Which of the following requires a device driver? (GATE CS 2001)
a) Register
b) Cache
c) Main memory
d) Disk

Answer: (d)

2. Automata Theory | Set 1

Following questions have been asked in GATE CS exam.

1. Let S and T be language over ={a,b} represented by the regular expressions (a+b*)* and (a+b)*, respectively. Which of the following is true? (GATE CS 2000)

(a) ScT (S is a subset of T)
(b) TcS (T is a subset of S)
(c) S=T
(d) SnT=Ø

Answer: (c).

2. Let L denotes the language generated by the grammar S – OSO/00. Which of the following is true? (GATE CS 2000)
(a) L = O
(b) L is regular but not O
(c) L is context free but not regular
(d) L is not context free

Answer: (b)
Explanation: Please note that grammar itself is not regular but language L is regular as L can be represented using a regular grammar, for example S -> S00/00.
References:
http://en.wikipedia.org/wiki/Regular_grammar

3. Consider the following two statements:
S1: { 0^2n |n >= l} is a regu1ar language
S2: { 0^m 0^n 0^(m+n) l m >= 1 and n >= 2} is a regu1ar language
Which of the following statements is correct? (GATE CS 2001)
a) Only S1 is correct
b) Only S2 is correct
c) Both S1 and S2 are correct
d) None of S1 and S2 is correct

Answer: (c)
Explanation:
S1 can be written as (00)^n where n >= 1. And S2 can be written as (00)^(m+n) where m >=2 and n >= 1. S2 can be further reduced to (00)^x where x >= 3.
We can easily write regular grammars for both S1 and S2.
G1 -> G100/00 (For S1)
G2 -> G200/000000 (For S2)

4. Which of the following statements in true? (GATE CS 2001)
(a) If a language is context free it can always be accepted by a deterministic push-down automaton

(b) The union of two context free languages is context free
(c) The intersection of two context free languages is context free
(d) The complement of a context free language is context free

Answer: (b)
Explanation:
Context-free languages are closed under the following operations. That is, if L and P are context-free languages and D is a regular language, the following languages are context-free as well:
• the Kleene star L * of L
• the image Ø(L) of L under a homomorphism Ø
• the concatenation of L and P
• the union of L and P
• the intersection of L with a regular language D (L n D).
Context-free languages are not closed under complement, intersection, or difference.
Why a) is not true?
The language recognized by deterministic pushdown automaton is deterministic context free language. Not all context-free languages are deterministic. This is unlike the situation for deterministic finite automata, which are also a subset of the nondeterministic finite automata but can recognize the same class of languages (as demonstrated by the subset construction).
References:
http://en.wikipedia.org/wiki/Context-free_language
http://en.wikipedia.org/wiki/Deterministic_pushdown_automaton

5. Given an arbitrary non-deterministic finite automaton (NFA) with N states, the maximum number of states in an equivalent minimized DFA is at least. (GATE CS 2001)
(a) N^2
(b) 2^N
(c) 2N
(d) N!

Answer: (b)
References:
http://en.wikipedia.org/wiki/Powerset_construction

3. Operating Systems | Set 2

Following questions have been asked in GATE CS exam.

1. Consider a machine with 64 MB physical memory and a 32-bit virtual address space. If the page size is 4KB, what is the approximate size of the page table? (GATE 2001)

(a) 16 MB
(b) 8 MB
(c) 2 MB
(d) 24 MB

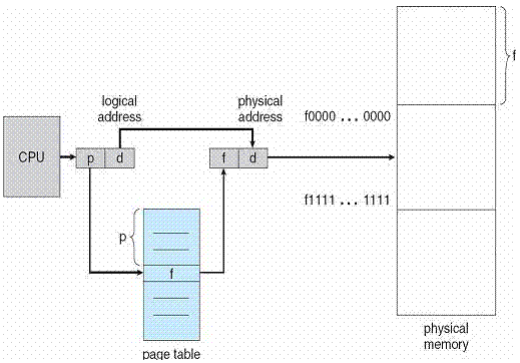
Answer: (c)
Explanation:
A page entry is used to get address of physical memory. Here we assume that single level of Paging is happening. So the resulting page table will contain entries for all the pages of the Virtual address space.

```
Number of entries in page table =
                    (virtual address space size)/(page size)
```

Using above formula we can say that there will be 2^(32-12) = 2^20 entries in page table.
No. of bits required to address the 64MB Physical memory = 26.
So there will be 2^(26-12) = 2^14 page frames in the physical memory. And page table needs to store the address of all these 2^14 page frames. Therefore, each page table entry will contain 14 bits address of the page frame and 1 bit for valid-invalid bit.
Since memory is byte addressable. So we take that each page table entry is 16 bits i.e. 2 bytes long.

```
Size of page table =
  (total number of page table entries) *(size of a page table entry)
   = (2^20 *2) = 2MB
```

For the clarity of the concept, please see the following figure. As per our question, here p = 20, d = 12 and f = 14.



2. Consider Peterson's algorithm for mutual exclusion between two concurrent

processes i and j. The program executed by process is shown below.

```
repeat
   flag [i] = true;
   turn = j;
   while ( P ) do no-op;
   Enter critical section, perform actions, then exit critical
   section
   flag [ i ] = false;
   Perform other non-critical section actions.
until false;
```

For the program to guarantee mutual exclusion, the predicate P in the while loop should be (GATE 2001)

a) flag [j] = true and turn = i
b) flag [j] = true and turn = j
c) flag [i] = true and turn = j
d) flag [i] = true and turn = i

Answer: (b)
Basically, Peterson's algorithm provides guaranteed mutual exclusion by using the two following constructs – *flag[]* and *turn*. *flag[]* controls that the willingness of a process to be entered in critical section. While *turn* controls the process that is allowed to be entered in critical section. So by replacing P with the following,

flag [j] = true and turn = j

process i will not enter critical section if process j wants to enter critical section and it is process j's *turn* to enter critical section. The same concept can be extended for more than two processes. For details, refer the following.
References:
http://en.wikipedia.org/wiki/Peterson%27s_algorithm

3 More than one word are put in one cache block to (GATE 2001)
(a) exploit the temporal locality of reference in a program
(b) exploit the spatial locality of reference in a program
(c) reduce the miss penalty
(d) none of the above

Answer: (b)
Temporal locality refers to the reuse of specific data and/or resources within relatively small time durations. Spatial locality refers to the use of data elements within relatively close storage locations.

To exploit the spatial locality, more than one word are put into cache block.
References:
http://en.wikipedia.org/wiki/Locality_of_reference

4. Which of the following statements is false? (GATE 2001)
a) Virtual memory implements the translation of a program's address space into physical memory address space
b) Virtual memory allows each program to exceed the size of the primary memory
c) Virtual memory increases the degree of multiprogramming
d) Virtual memory reduces the context switching overhead

Answer: (d)
In a system with virtual memory context switch includes extra overhead in switching of address spaces.
References:
http://www.itee.adfa.edu.au/~spike/CSA2/Lectures00/lecture.vm.htm

5. Consider a set of n tasks with known runtimes r1, r2, … rn to be run on a uniprocessor machine. Which of the following processor scheduling algorithms will result in the maximum throughput? (GATE 2001)
(a) Round-Robin
(b) Shortest-Job-First
(c) Highest-Response-Ratio-Next
(d) First-Come-First-Served

Answer: (b)

4. Operating Systems | Set 3

Following questions have been asked in GATE CS exam.

1. Suppose the time to service a page fault is on the average 10 milliseconds, while a memory access takes 1 microsecond. Then a 99.99% hit ratio results in average memory access time of (GATE CS 2000)
(a) 1.9999 milliseconds
(b) 1 millisecond
(c) 9.999 microseconds
(d) 1.9999 microseconds

Answer: (d)

Explanation:

```
Average memory access time =
      [(% of page miss)*(time to service a page fault) +
                  (% of page hit)*(memory access time)]/100
```

So, average memory access time in microseconds is.
(99.99*1 + 0.01*10*1000)/100 = (99.99+100)/1000 = 199.99/1000 =1.9999 µs

2. Which of the following need not necessarily be saved on a context switch between processes? (GATE CS 2000)
(a) General purpose registers
(b) Translation look-aside buffer
(c) Program counter
(d) All of the above

Answer: (b)
Explanation:
In a process context switch, the state of the first process must be saved somehow, so that, when the scheduler gets back to the execution of the first process, it can restore this state and continue.

The state of the process includes all the registers that the process may be using, especially the program counter, plus any other operating system specific data that may be necessary.

A Translation lookaside buffer (TLB) is a CPU cache that memory management hardware uses to improve virtual address translation speed. A TLB has a fixed number of slots that contain page table entries, which map virtual addresses to physical addresses. On a context switch, some TLB entries can become invalid, since the virtual-to-physical mapping is different. The simplest strategy to deal with this is to completely flush the TLB.
References:
http://en.wikipedia.org/wiki/Context_switch
http://en.wikipedia.org/wiki/Translation_lookaside_buffer#Context_switch

3. Where does the swap space reside ? (GATE 2001)
(a) RAM
(b) Disk
(c) ROM
(d) On-chip cache
Answer: (b)
Explanation:
Swap space is an area on disk that temporarily holds a process memory image. When physical memory demand is sufficiently low, process memory images are brought back

into physical memory from the swap area. Having sufficient swap space enables the system to keep some physical memory free at all times.

References:

http://docs.hp.com/en/B2355-90672/ch06s02.html

4. Which of the following does not interrupt a running process? (GATE CS 2001)

(a) A device
(b) Timer
(c) Scheduler process
(d) Power failure

Answer: (c)

Explanation:

Scheduler process doesn't interrupt any process, it's Job is to select the processes for following three purposes.

Long-term scheduler(or job scheduler) –selects which processes should be brought into the ready queue

Short-term scheduler(or CPU scheduler) –selects which process should be executed next and allocates CPU.

Mid-term Scheduler (Swapper)- present in all systems with virtual memory, temporarily removes processes from main memory and places them on secondary memory (such as a disk drive) or vice versa. The mid-term scheduler may decide to swap out a process which has not been active for some time, or a process which has a low priority, or a process which is page faulting frequently, or a process which is taking up a large amount of memory in order to free up main memory for other processes, swapping the process back in later when more memory is available, or when the process has been unblocked and is no longer waiting for a resource.

5. Which of the following scheduling algorithms is non-preemptive? (GATE CS 2002)

a) Round Robin
b) First-In First-Out
c) Multilevel Queue Scheduling
d) Multilevel Queue Scheduling with Feedback

Answer: (b)

5. Database Management Systems | Set 1

Following questions have been asked in GATE CS exam.

1. Given the relations

employee (name, salary, deptno) and
department (deptno, deptname, address)

Which of the following queries cannot be expressed using the basic relational algebra operations (U, -, x, π, σ, p)? (GATE CS 2000)
(a) Department address of every employee
(b) Employees whose name is the same as their department name
(c) The sum of all employees' salaries
(d) All employees of a given department

Answer: (c)

Explanation:
The six basic operators of relational algebra are the selection(σ), the projection(π), the Cartesian product (x) (also called the cross product or cross join), the set union (U), the set difference (-), and the rename (p). These six operators are fundamental in the sense that none of them can be omitted without losing expressive power. Many other operators have been defined in terms of these six. Among the most important are set intersection, division, and the natural join, but aggregation is not possible with these basic relational algebra operations. So, we cannot run sum of all employees' salaries with the six operations.

References:
http://en.wikipedia.org/wiki/Relational_algebra
http://faculty.ksu.edu.sa/zitouni/203%20Haseb%20%20Lecture%20Notes/Relional%20Algebra.

2. Given the following relation instance.

x	y	z
1	4	2
1	5	3
1	6	3
3	2	2

Which of the following functional dependencies are satisfied by the instance? (GATE CS 2000)
(a) XY -> Z and Z -> Y
(b) YZ -> X and Y -> Z
(c) YZ -> X and X -> Z
(d) XZ -> Y and Y -> X

Answer: (b)

Explanation:
A functional dependency (FD) is a constraint between two sets of attributes in a relation from a database. A FD X->Y require that the value of X uniquely determines the value of Y where X and Y are set of attributes. FD is a generalization of the notion of a key.

Given that X, Y, and Z are sets of attributes in a relation R, one can derive several properties of functional dependencies. Among the most important are Armstrong's axioms, which are used in database normalization:

```
* Subset Property (Axiom of Reflexivity): If Y is a subset of X, then X ? Y
* Augmentation (Axiom of Augmentation): If X -> Y, then XZ -> YZ
* Transitivity (Axiom of Transitivity): If X -> Y and Y -> Z, then X -> Z
```

From these rules, we can derive these secondary rules:

```
* Union: If X -> Y and X -> Z, then X -> YZ
* Decomposition: If X -> YZ, then X -> Y and X -> Z
* Pseudotransitivity: If X -> Y and YZ -> W, then XZ -> W
```

In the above question, Y uniquely determines X and Z, for a given value of Y you can easily find out values of X and Z.
So, Y -> X and Y -> Z hold for above schema.
From rule of augmentation we can say YZ->X. If we understand the notion of FD, we don't need to apply axioms to find out which option is true, just by looking at the schema and options we can say that (b) is true.

References:
http://www.cse.iitb.ac.in/~sudarsha/db-book/slide-dir/ch7.pdf
http://en.wikipedia.org/wiki/Functional_dependency

3. Given relations r(w, x) and s(y, z), the result of
select distinct w, x
from r, s
is guaranteed to be same as r, provided (GATE CS 2000)
(a) r has no duplicates and s is non-empty
(b) r and s have no duplicates
(c) s has no duplicates and r is non-empty
(d) r and s have the same number of tuples

Answer: (a)

Explanation:

The query selects all attributes of r. Since we have distinct in query, result can be equal to r only if r doesn't have duplicates.

If we do not give any attribute on which we want to join two tables, then the queries like above become equivalent to Cartesian product. Cartisian product of two sets will be empty if any of the two sets is empty. So, s should have atleast one record to get all rows of r.

4. In SQL, relations can contain null values, and comparisons with null values are treated as unknown. Suppose all comparisons with a null value are treated as false. Which of the
following pairs is not equivalent? (GATE CS 2000)
(a) x = 5, not (not (x = 5)
(b) x = 5, x > 4 and x < 6, where x is an integer
(c) x < 5, not(x = 5)
(d) None of the above

Answer (c)

Explanation:
It doesn't need much explanation. For all values smaller than 5, x < 5 will always be true but x = 5 will be false.

5. Consider a schema R(A, B, C, D) and functional dependencies A -> B and C -> D. Then the decomposition of R into R1 (A, B) and R2(C, D) is (GATE CS 2001)
a) dependency preserving and loss less join
b) loss less join but not dependency preserving
c) dependency preserving but not loss less join
d) not dependency preserving and not loss less join

Answer: (c)

Explanation:
Dependency Preserving Decomposition:
Decomposition of R into R1 and R2 is a dependency preserving decomposition if closure of functional dependencies after decomposition is same as closure of of FDs before decomposition.
A simple way is to just check whether we can derive all the original FDs from the FDs present after decomposition.

In the above question R(A, B, C, D) is decomposed into R1 (A, B) and R2(C, D) and there are only two FDs A -> B and C -> D. So, the decomposition is dependency preserving

Lossless-Join Decomposition:
Decomposition of R into R1 and R2 is a lossless-join decomposition if at least one of the following functional dependencies are in F+ (Closure of functional dependencies)

```
   R1  R2  R1
  OR
   R1  R2  R2
```

In the above question R(A, B, C, D) is decomposed into R1 (A, B) and R2(C, D), and R1 ∩ R2 is empty. So, the decomposition is not lossless.

References:
http://www.cs.sfu.ca/CC/354/han/materia/notes/354notes-chapter6/node1.html

6. C Language | Set 1

Following questions have been asked in GATE CS exam.

1. Consider the following three C functions :

```
[PI] int * g (void)
{
  int x = 10;
  return (&x);
}

[P2] int * g (void)
{
  int * px;
  *px = 10;
  return px;
}

[P3] int *g (void)
{
  int *px;
  px = (int *) malloc (sizeof(int));
  *px = 10;
  return px;
}
```

Which of the above three functions are likely to cause problems with pointers? (GATE 2001)
(a) Only P3
(b) Only P1 and P3
(c) Only P1 and P2
(d) P1, P2 and P3

Answer: (c)

Eplaination: In P1, pointer variable x is a local variable to g(), and g() returns pointer to this variable. x may vanish after g() has returned as x exists on stack. So, &x may become invalid.
In P2, pointer variable px is being assigned a value without allocating memory to it.
P3 works perfectly fine. Memory is allocated to pointer variable px using malloc(). So, px exists on heap, it's existence will remain in memory even after return of g() as it is on heap.

2. The value of j at the end of the execution of the following C program. (GATE CS 2000)

```
int incr (int i)
{
   static int count = 0;
   count = count + i;
   return (count);
}
main ()
{
   int i,j;
   for (i = 0; i <=4; i++)
      j = incr(i);
}
```

(a) 10
(b) 4
(c) 6
(d) 7

Answer (a)
Eplaination: count is static variable in incr(). Statement static int count = 0 will assign count to 0 only in first call. Other calls to this function will take the old values of count.
Count will become 0 after the call incr(0)
Count will become 1 after the call incr(1)
Count will become 3 after the call incr(2)
Count will become 6 after the call incr(3)
Count will become 10 after the call incr(4)

3. Consider the following C declaration

```
struct {
    short s [5]
    union {
         float y;
         long z;
    }u;
} t;
```

Assume that objects of the type short, float and long occupy 2 bytes, 4 bytes and 8

bytes, respectively. The memory requirement for variable t, ignoring alignment considerations, is (GATE CS 2000)
(a) 22 bytes
(b) 14 bytes
(c) 18 bytes
(d) 10 bytes

Answer: (c)
Explanation: Short array s[5] will take 10 bytes as size of short is 2 bytes. Since u is a union, memory allocated to u will be max of float y(4 bytes) and long z(8 bytes). So, total size will be 18 bytes (10 + 8).

4. The number of tokens in the following C statement.

```
printf("i = %d, &i = %x", i, &i);
```

is (GATE 2000)
(a) 3
(b) 26
(c) 10
(d) 21

Answer (c)
Explanation: In a C source program, the basic element recognized by the compiler is the "token." A token is source-program text that the compiler does not break down into component elements.
There are 6 types of C tokens : identifiers, keywords, constants, operators, string literals and other separators. There are total 10 tokens in the above printf statement.

5. The following C declarations

```
struct node
{
   int i;
   float j;
};
struct node *s[10] ;
```

define s to be (GATE CS 2000)

(a) An array, each element of which is a pointer to a structure of type node
(b) A structure of 2 fields, each field being a pointer to an array of 10 elements
(c) A structure of 3 fields: an integer, a float, and an array of 10 elements
(d) An array, each element of which is a structure of type node.

Answer: (a)

7. Compiler Theory | Set 1

Following questions have been asked in GATE CS exam.

1. Which of the following derivations does a top-down parser use while parsing an input string? The input is assumed to be scanned in left to right order (GATE CS 2000).
(a) Leftmost derivation
(b) Leftmost derivation traced out in reverse
(c) Rightmost derivation
(d) Rightmost derivation traced out in reverse

Answer (a)

Top-down parsing (LL)
In top down parsing, we just start with the start symbol and compare the right side of the different productions against the first piece of input to see which of the productions should be used.

A top down parser is called LL parser because it parses the input from **L**eft to right, and constructs a **L**eftmost derivation of the sentence.

Algorithm (Top Down Parsing)

```
a) In the current string, choose leftmost nonterminal.
b) Choose a production for the chosen nonterminal.
c) In the string, replace the nonterminal by the right-hand-side
   of the rule.
d) Repeat until no more nonterminals.
```

LL grammars are often classified by numbers, such as LL(1), LL(0) and so on. The number in the parenthesis tells the maximum number of terminals we may have to look at at a time to choose the right production at any point in the grammar.

The most common (and useful) kind of LL grammar is LL(1) where you can always choose the right production by looking at only the first terminal on the input at any given time. With LL(2) you have to look at two symbols, and so on. There exist grammars that are not LL(k) grammars for any fixed value of k at all, and they are sadly quite common.

Let us see an example of top down parsing for following grammar. Let input string be ax.

```
S -> Ax
A -> a
A -> b
```

An LL(1) parser starts with S and asks “which production should I attempt?” Naturally, it predicts the only alternative of S. From there it tries to match A by calling method A (in a recursive-descent parser). Lookahead a predicts production

```
A -> a
```

The parser matches a, returns to S and matches x. Done. The derivation tree is:

```
   S
  / \
A   x
|
a
```

References:
http://www.garshol.priv.no/download/text/bnf.html
http://en.wikipedia.org/wiki/Top-down_parsing
http://www.cs.wm.edu/~noonan/animations/lderive.html
http://en.wikipedia.org/wiki/LL_parser

2. The process of assigning load addresses to the various parts of the program and adjusting the code and data in the program to reflect the assigned addresses is called (GATE CS 2001)
a) Assembly
b) Parsing
c) Relocation
d) Symbol resolution

Answer: (c)
Relocation is the process of replacing symbolic references or names of libraries with actual usable addresses in memory before running a program. It is typically done by the linker during compilation (at compile time), although it can be done at runtime by a relocating loader. Compilers or assemblers typically generate the executable with zero as the lower-most starting address. Before the execution of object code, these addresses should be adjusted so that they denote the correct runtime addresses.

Relocation is typically done in two steps:
1. Each object code has various sections like code, data, .bss etc. To combine all the objects to a single executable, the linker merges all sections of similar type into a single section of that type. The linker then assigns runtime addresses to each section and each symbol. At this point, the code (functions) and data (global variables) will have unique runtime addresses.
2. Each section refers to one or more symbols which should be modified so that they point to the correct runtime addresses.

References:
http://en.wikipedia.org/wiki/Relocation_(computer_science)

3. Which of the following statements is false? (GATE CS 2001)
a) An unambiguous grammar has same leftmost and rightmost derivation
b) An LL(1) parser is a top-down parser
c) LALR is more powerful than SLR
d) An ambiguous grammar can never be LR(k) for any k

Answer: (a)
If a grammar has more than one leftmost (or rightmost) derivation for a single sentential form, the grammar is ambiguous. The leftmost and rightmost derivations for a sentential form may differ, even in an unambiguous grammar

4. Which of the following grammar rules violate the requirements of an operator grammar? P, Q, R are nonterminals, and r,s,t are terminals (GATE CS 2004).
(i) P -> QR
(ii) P -> QsR
(iii) P -> ϵ
(iV) P -> QtRr

a) (i) only
b) (i) and (iii) only
c) (ii) and (iii) only
d) (iii) and (iv) only

Answer: (b)
Explanation:
An *operator precedence parser* is a bottom-up parser that interprets an operator-precedence grammar. For example, most calculators use operator precedence parsers to convert from the human-readable infix notation with order of operations format into an internally optimized computer-readable format like Reverse Polish notation (RPN).

An *operator precedence grammar* is a kind of context-free grammar that can be parsed with an operator-precedence parser. It has the property that no production has either an empty (ϵ) right-hand side or two adjacent nonterminals in its right-hand side. These properties allow the terminals of the grammar to be described by a precedence relation, and the a parser that exploits that relation is considerably simpler than more general-purpose parsers such as LALR parsers.

References:
http://en.wikipedia.org/wiki/Operator-precedence_grammar
http://en.wikipedia.org/wiki/Operator-precedence_parser

5. Consider the grammar with the following translation rules and E as the start symbol.
E -> E1 #T {E.value = E1.value * T.value}
| T {E.value = T.value}
T -> T1 & F {T.value = T1.value + F.value}
|F {T.value= F.value}
F -> num {F.value = num.value}

Compute E.value for the root of the parse tree for the expression:2 # 3 & 5 # 6 &4.
(GATE CS 2004)
a) 200
b) 180
c) 160
d) 40

Answer: (c)
Explanation:
We can calculate the value by constructing the parse tree for the expression 2 # 3 & 5 # 6 &4.

Alternatively, we can calculate by considering following precedence and associativity rules.
Precedence in a grammar is enforced by making sure that a production rule with higher precedence operator will never produce an expression with operator with lower precedence.
In the given grammar '&' has higher precedence than '#'.

Left associativity for operator * in a grammar is enforced by making sure that for a production rule like S -> S1 * S2 in grammar, S2 should never produce an expression with *. On the other hand, to ensure right associativity, S1 should never produce an expression with *.
In the given grammar, both '#' and & are left-associative.

So expression 2 # 3 & 5 # 6 &4 will become
((2 # (3 & 5)) # (6 & 4))
Let us apply translation rules, we get
((2 * (3 + 5)) * (6 + 4)) = 160.

8. Data Structures and Algorithms | Set 1

Following questions have been asked in GATE CS exam

1. Let LASTPOST, LASTIN and LASTPRE denote the last vertex visited in a

postorder, inorder and preorder traversal. Respectively, of a complete binary tree. Which of the following is always true? (GATE CS 2000)
(a) LASTIN = LASTPOST
(b) LASTIN = LASTPRE
(c) LASTPRE = LASTPOST
(d) None of the above

Answer (d)

It is given that the given tree is complete binary tree. For a complete binary tree, the last visited node will always be same for inorder and preorder traversal. None of the above is true even for a complete binary tree.

The option (a) is incorrect because the last node visited in Inorder traversal is right child and last node visited in Postorder traversal is root.

The option (c) is incorrect because the last node visited in Preorder traversal is right child and last node visited in Postorder traversal is root.

For option (b), see the following counter example. Thanks to Hunaif Muhammed for providing the correct explanation.

```
     1
   /    \
  2      3
 / \    /
4   5  6

Inorder traversal is 4 2 5 1 6 3
Preorder traversal is 1 2 4 5 3 6
```

2. The most appropriate matching for the following pairs

X: depth first search	**1: heap**
Y: breadth-first search	**2: queue**
Z: sorting	**3: stack**

is (GATE CS 2000):
(a) X—1 Y—2 Z-3
(b) X—3 Y—1 Z-2
(c) X—3 Y—2 Z-1
(d) X—2 Y—3 Z-1

Answer: (c)
Stack is used for Depth first Search
Queue is used for Breadth First Search

Heap is used for sorting

3. Consider the following nested representation of binary trees: (X Y Z) indicates Y and Z are the left and right sub stress, respectively, of node X. Note that Y and Z may be NULL, or further nested. Which of the following represents a valid binary tree?
(a) (1 2 (4 5 6 7))
(b) (1 (2 3 4) 5 6) 7)
(c) (1 (2 3 4)(5 6 7))
(d) (1 (2 3 NULL) (4 5))

Answer (c)

4. Let s be a sorted array of n integers. Let t(n) denote the time taken for the most efficient algorithm to determined if there are two elements with sum less than 1000 in s. which of the following statements is true? (GATE CS 2000)
a) t (n) is 0 (1)
b) n < t (n) < n$log_2 n$
c) n log 2 n < t (n) < $\binom{n}{2}$
d) t (n) = $\binom{n}{2}$

Answer (a)
Let array be sorted in ascending order, if sum of first two elements is less than 1000 then there are two elements with sum less than 1000 otherwise not. For array sorted in descending order we need to check last two elements. For an array data structure, number of operations are fixed in both the cases and not dependent on n, complexity is O(1)

5. B+ trees are preferred to binary trees in databases because (GATE CS 2000)
(a) Disk capacities are greater than memory capacities
(b) Disk access is much slower than memory access
(c) Disk data transfer rates are much less than memory data transfer rates
(d) Disks are more reliable than memory

Answer (b)
Disk access is slow and B+ Tree provide search in less number of disk hits. This is primarily because unlike binary seach trees, B+ trees have very high fanout (typically on the order of 100 or more), which reduces the number of I/O operations required to find an element in the tree.

9. C Language | Set 2

Following questions have been asked in GATE CS exam.

1. Consider the following C program segment:

```
char p[20];
char *s = "string";
int length = strlen(s);
int i;
for (i = 0; i < length; i++)
     p[i] = s[length – i];
printf("%s",p);
```

The output of the program is (GATE CS 2004)

a) gnirts

b) gnirt

c) string

d) no output is printed

Answer(d)

Let us consider below line inside the for loop

p[i] = s[length — i];

For i = 0, p[i] will be s[6 — 0] and s[6] is '\0'

So p[0] becomes '\0'. It doesn't matter what comes in p[1], p[2]….. as P[0] will not change for i >0. Nothing is printed if we print a string with first character '\0'

2. Consider the following C function

```
void swap (int a, int b)
{
   int temp;
   temp = a;
   a = b;
   b = temp;
}
```

In order to exchange the values of two variables x and y. (GATE CS 2004)

a) call swap (x, y)

b) call swap (&x, &y)

c) swap (x,y) cannot be used as it does not return any value

d) swap (x,y) cannot be used as the parameters are passed by value

Answer(d)

Why a, b and c are incorrect?

a) call swap (x, y) will not cause any effect on x and y as parameters are passed by

value.
b) call swap (&x, &y) will no work as function swap() expects values not addresses (or pointers).
c) swap (x, y) cannot be used but reason given is not correct.

3. Consider the following C function:

```
int f(int n)
{
   static int i = 1;
   if (n >= 5)
      return n;
   n = n+i;
   i++;
   return f(n);
}
```

The value returned by f(1) is (GATE CS 2004)
a) 5
b) 6
c) 7
d) 8

Answer (c)
Since i is static, first line of f() is executed only once.

```
Execution of f(1)
    i = 1
    n = 2
    i = 2
 Call f(2)
    i = 2
    n = 4
    i = 3
 Call f(4)
   i = 3
   n = 7
   i = 4
 Call f(7)
  since n >= 5 return n(7)
```

4. Consider the following program fragment for reversing the digits in a given integer to obtain a new integer. Let n = D1D2…Dm

```
int n, rev;
rev = 0;
while (n > 0)
{
   rev = rev*10 + n%10;
   n = n/10;
}
```

The loop invariant condition at the end of the ith iteration is:(GATE CS 2004)

a) n = D1D2….Dm-i and rev = DmDm-1…Dm-i+1
b) n = Dm-i+1…Dm-1Dm and rev = Dm-1….D2D1
c) n ≠ rev
d) n = D1D2….Dm and rev = DmDm-1…D2D1

Answer (a)

5. Consider the following C program

```
main()
{
   int x, y, m, n;
   scanf ("%d %d", &x, &y);
   /* x > 0 and y > 0 */
   m = x; n = y;
   while (m != n)
   {
      if(m>n)
         m = m - n;
      else
         n = n - m;
   }
   printf("%d", n);
}
```

The program computes (GATE CS 2004)
a) x + y using repeated subtraction
b) x mod y using repeated subtraction
c) the greatest common divisor of x and y
d) the least common multiple of x and y

Answer(c)
This is an implementation of Euclid's algorithm to find GCD

10. Data Structures and Algorithms | Set 2

Following questions have been asked in GATE CS exam.

1. Consider the function f defined below.

```
struct item
{
  int data;
  struct item * next;
};

int f(struct item *p)
{
  return (
          (p == NULL) ||
          (p->next == NULL) ||
          (( P->data &lt;= p->next->data) &amp;&amp; f(p->next))
         );
}
```

For a given linked list p, the function f returns 1 if and only if (GATE CS 2003)
a) the list is empty or has exactly one element
b) the elements in the list are sorted in non-decreasing order of data value
c) the elements in the list are sorted in non-increasing order of data value
d) not all elements in the list have the same data value.

Answer (b)
The function f() works as follows
1) If linked list is empty return 1
2) Else If linked list has only one element return 1
3) Else if node->data is smaller than equal to node->next->data and same thing holds for rest of the list then return 1
4) Else return 0

2. Consider the label sequences obtained by the following pairs of traversals on a labeled binary tree. Which of these pairs identify a tree uniquely (GATE CS 2004)?
i) preorder and postorder
ii) inorder and postorder
iii) preorder and inorder
iv) level order and postorder

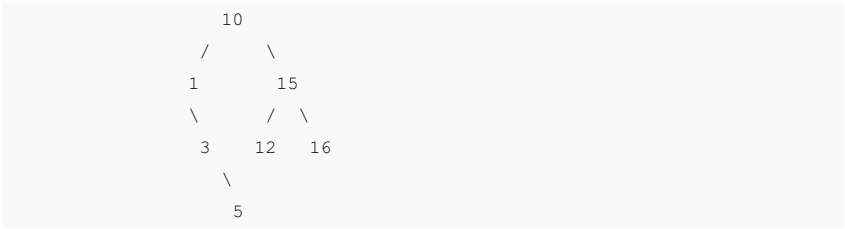
a) (i) only
b) (ii), (iii)
c) (iii) only
d) (iv) only

Answer (b)
Please see http://geeksforgeeks.org/?p=657 for explanation.

3. The following numbers are inserted into an empty binary search tree in the given order: 10, 1, 3, 5, 15, 12, 16. What is the height of the binary search tree (the height is the maximum distance of a leaf node from the root)? (GATE CS 2004)
a) 2
b) 3

c) 4
d) 6

Answer(b)
Constructed binary search tree will be..



4. A data structure is required for storing a set of integers such that each of the following operations can be done in (log n) time, where n is the number of elements in the set.

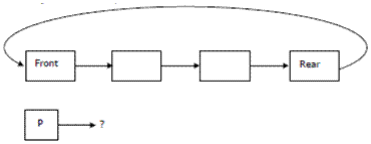
```
o Delection of the smallest element
o Insertion of an element if it is not already present in the set
```

Which of the following data structures can be used for this purpose?
(a) A heap can be used but not a balanced binary search tree
(b) A balanced binary search tree can be used but not a heap
(c) Both balanced binary search tree and heap can be used
(d) Neither balanced binary search tree nor heap can be used

Answer(b)
A self-balancing balancing binary search tree containing n items allows the lookup, insertion, and removal of an item in O(log n) worst-case time. Since it's a self balancing BST, we can easily find out minimum element in O(logn) time which is always the leftmost element (See http://geeksforgeeks.org/?p=1333).

Since Heap is a balanced binary tree (or almost complete binary tree), insertion complexity for heap is O(logn). Also complexity to get minimum in a min heap is O(logn) because removal of root node causes a call to heapify (after removing the first element from the array) to maintain the heap tree property. But a heap cannot be used for the above purpose as the question says – insert an element if it is not already present. For a heap, we cannot find out in O(logn) if an element is present or not. Thanks to game for providing the correct solution.

5. A circularly linked list is used to represent a Queue. A single variable p is used to access the Queue. To which node should p point such that both the operations enQueue and deQueue can be performed in constant time? (GATE 2004)



a) rear node
b) front node
c) not possible with a single pointer
d) node next to front

Answer(a)
Answer is not “(b) front node”, as we can not get rear from front in O(1), but if p is rear we can implement both enQueue and deQueue in O(1) because from rear we can get front in O(1). Below are sample functions. Note that these functions are just sample are not working. Code to handle base cases is missing.

```
/* p is pointer to address of rear (double pointer).  This function ad
   node after rear and updates rear which is *p to point to new node
void  enQueue(struct node **p, struct node *new_node)
{
    /* Missing code to handle base cases like *p is NULL */

     new_node->next = (*p)->next;
     (*p)->next = new_node;
     (*p) = new_node /* new is now rear */
     /* Note that p is again front and  p->next is rear */
 }

/* p is pointer to rear. This function removes the front element and
    returns the new front */
struct node *deQueue(struct node *p)
{
    /* Missing code to handle base cases like p is NULL,
        p->next is NULL,...  etc */

    struct node *temp = p->next->next;
    p->next = p->next->next;
    return temp;
    /* Note that p is again front and  p->next is rear */
}
```

11. Data Structures and Algorithms | Set 3

Following questions have asked in GATE CS exam.

1. Suppose you are given an array s[1…n] and a procedure reverse (s,i,j) which reverses the order of elements in a between positions i and j (both inclusive). What does the following sequence

```
do, where 1 < k <= n:
  reverse (s, 1, k);
  reverse (s, k + 1, n);
  reverse (s, 1, n);
```

(GATE CS 2000)
(a) Rotates s left by k positions
(b) Leaves s unchanged
(c) Reverses all elements of s
(d) None of the above

Answer: (a)
Effect of the above 3 reversals for any k is equivalent to left rotation of the array of size n by k. Please see this post for details.
If we rotate an array n times for k = 1 to n, we get the same array back.

2. The best data structure to check whether an arithmetic expression has balanced parentheses is a (GATE CS 2004)
a) queue
b) stack
c) tree
d) list

Answer(b)
There are three types of parentheses [ ] { } (). Below is an arbit c code segment which has parentheses of all three types.

```
void func(int c, int a[])
{
   return  ((c +2)  + arr[(c-2)]) ;
}
```

Stack is a straightforward choice for checking if left and right parentheses are balanced. Here is a algorithm to do the same.

```
/*Return 1 if expression has balanced parentheses */
bool areParenthesesBalanced(expression )
{
   for each character in expression
   {
      if(character == '(' || character == '{' || character == '[')
        push(stack, character);
      if(character == ')' || character == '}' || character == ']')
      {
         if(isEmpty(stack))
           return 0; /*We are seeing a right parenthesis
                       without a left pair*/

         /* Pop the top element from stack, if it is not a pair
             bracket of character then there is a mismatch.
             This will happen for expressions like {(}) */
         else if (! isMatchingPair(pop(stack), character) )
           return 0;
      }
   }

   if(isEmpty(stack))
     return 1; /*balanced*/
   else
     return 0;  /*not balanced*/
} /* End of function to check parentheses */

/* Returns 1 if character1 and character2 are matching left
   and right parentheses */
bool isMatchingPair(character1, character2)
{
   if(character1 == '(' && character2 == ')')
     return 1;
   else If(character1 == '{' && character2 == '}')
     return 1;
   else If(character1 == '[' && character2 == ']')
     return 1;
   else
     return 0;
}
```

3. Level order traversal of a rooted tree can be done by starting from the root and performing (GATE CS 2004)

a) preorder traversal

b) in-order traversal

c) depth first search

d) breadth first search

Answer(d)

See this post for details

4. Given the following input (4322, 1334, 1471, 9679, 1989, 6171, 6173, 4199) and the hash function x mod 10, which of the following statements are true?

i. 9679, 1989, 4199 hash to the same value

ii. 1471, 6171 has to the same value

iii. All elements hash to the same value
iv. Each element hashes to a different value
(GATE CS 2004)

a) i only
b) ii only
c) i and ii only
d) iii or iv

Answer (c)

5. Postorder traversal of a given binary search tree, T produces the following sequence of keys
10, 9, 23, 22, 27, 25, 15, 50, 95, 60, 40, 29
Which one of the following sequences of keys can be the result of an in-order traversal of the tree T? (GATE CS 2005)
a) 9, 10, 15, 22, 23, 25, 27, 29, 40, 50, 60, 95
b) 9, 10, 15, 22, 40, 50, 60, 95, 23, 25, 27, 29
c) 29, 15, 9, 10, 25, 22, 23, 27, 40, 60, 50, 95
d) 95, 50, 60, 40, 27, 23, 22, 25, 10, 9, 15, 29

Answer (a)
Inorder traversal of a BST always gives elements in increasing order. Among all four options, a) is the only increasing order sequence.

12. Data Structures and Algorithms | Set 4

Following questions have been asked in GATE CS exam.

1. Consider the following C program segment

```
struct CellNode
{
  struct CelINode *leftchild;
  int element;
  struct CelINode *rightChild;
}

int Dosomething(struct CelINode *ptr)
{
    int value = 0;
    if (ptr != NULL)
    {
      if (ptr->leftChild != NULL)
        value = 1 + DoSomething(ptr->leftChild);
      if (ptr->rightChild != NULL)
        value = max(value, 1 + DoSomething(ptr->rightChild));
    }
    return (value);
}
```

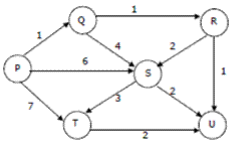
The value returned by the function DoSomething when a pointer to the root of a non-empty tree is passed as argument is (GATE CS 2004)

a) The number of leaf nodes in the tree
b) The number of nodes in the tree
c) The number of internal nodes in the tree
d) The height of the tree

Answer: (d)
Explanation: DoSomething() returns max(height of left child + 1, height of left child + 1). So given that pointer to root of tree is passed to DoSomething(), it will return height of the tree. Note that this implementation follows the convention where height of a single node is 0.

2. Suppose we run Dijkstra's single source shortest-path algorithm on the following edge weighted directed graph with vertex P as the source. In what order do the nodes get included into the set of vertices for which the shortest path distances are finalized? (GATE CS 2004)

a) P, Q, R, S, T, U
b) P, Q, R, U, S, T
c) P, Q, R, U, T, S
d) P, Q, T, R, U, S



Answer (b)

3. Suppose each set is represented as a linked list with elements in arbitrary order.

Which of the operations among union, intersection, membership, cardinality will be the slowest? (GATE CS 2004)

a) union only
b) intersection, membership
c) membership, cardinality
d) union, intersection

Answer (d)
Cardinality and membership are definitely not the slowest one. For cardinality, just count the number of nodes in a list. For membership, just traverse the list and look for a match

For getting intersection of L1 and L2, search for each element of L1 in L2 and print the elements we find in L2.

There can be many ways for getting union of L1 and L2. One of them is as follows
a) Print all the nodes of L1 and print only those which are not present in L2.
b) Print nodes of L2.

4. The time complexity of the following C function is (assume n > 0 (GATE CS 2004)

```
int recursive (mt n)
{
   if (n == 1)
     return (1);
   else
     return (recursive (n-1) + recursive (n-1));
}
```

a) 0(n)
b) 0(nlogn)
c) 0(n^2)
d) 0(2^n)

Answer(d)
Explanation:
Recursive expression for the above program will be.

```
T(n) = 2T(n-1) + c
T(1) = c1.
```

Let us solve it.

```
T(n) = 2(2T(n-2) + c) + c         = 4T(n-2) + 3c
T(n) = 8T(n-3) + 6c + c           =  8T(n-3) + 7c
T(n) = 16T(n-4) + 14c + c         =  16T(n-4) + 15c
.....................................................
.....................................................
T(n) = (2^(n-1))T(1) +  (2^(n-1) - 1)c
```

```
T(n) = O(2^n)
```

Please write comments if you find any of the above answers/explanations incorrect.

13. C Language | Set 3

Following questions have been asked in GATE CS exam.

1.Assume the following C variable declaration

```
int *A [10], B[10][10];
```

Of the following expressions
I A[2]
II A[2][3]
III B[1]
IV B[2][3]
which will not give compile-time errors if used as left hand sides of assignment statements in a C program (GATE CS 2003)?

a) I, II, and IV only
b) II, III, and IV only
c) II and IV only
d) IV only

Answer (a)
See below program

```
int main()
{
  int *A[10], B[10][10];
  int C[] = {12, 11, 13, 14};

  /* No problem with below statement as A[2] is a pointer
     and we are assigning a value to pointer */
  A[2] = C;

  /* No problem with below statement also as array style indexing
      can be done with pointers*/
  A[2][3] = 15;

  /* Simple assignment to an element of a 2D array*/
  B[2][3]  = 15;

  printf("%d %d", A[2][0], A[2][3]);
  getchar();
}
```

2. Consider the following declaration of a 'two-dimensional array in C:

```
char a[100][100];
```

Assuming that the main memory is byte-addressable and that the array is stored starting from memory address 0, the address of a[40][50] is (GATE CS 2002)
a) 4040
b) 4050
c) 5040
d) 5050

Answer(b)

```
Address of a[40][50] =
     Base address + 40*100*element_size + 50*element_size
     0 + 4000*1 + 50*1
     4050
```

3. The C language is. (GATE CS 2002)
a) A context free language
b) A context sensitive language
c) A regular language
d) Parsable fully only by a Turing machine

Answer (a)
Most programming languages including C, C++, Java and Pascal can be approximated by context-free grammar and compilers for them have been developed based on properties of context-free languages.

References:
http://acm.pku.edu.cn/JudgeOnline/problem?id=3220
http://www.cs.odu.edu/~toida/nerzic/390teched/cfl/cfg.html

4 The most appropriate matching for the following pairs (GATE CS 2000)

```
X: m=malloc(5); m= NULL;        1: using dangling pointers
Y: free(n); n->value=5;         2: using uninitialized pointers
Z: char *p; *p = 'a';           3. lost memory is:
```

(a) X—1 Y—3 Z-2
(b) X—2 Y—1 Z-3
(C) X—3 Y—2 Z-1
(d) X—3 Y—1 Z-2

Answer (d)
X -> A pointer is assigned to NULL without freeing memory so a clear example of memory leak
Y -> Trying to retrieve value after freeing it so dangling pointer.
Z -> Using uninitialized pointers

5. Consider the following C-program:

```
void foo(int n, int sum)
{
  int k = 0, j = 0;
  if (n == 0) return;
    k = n % 10;
  j = n / 10;
  sum = sum + k;
  foo (j, sum);
  printf ("%d,", k);
}

int main ()
{
  int a = 2048, sum = 0;
  foo (a, sum);
  printf ("%d\n", sum);

  getchar();
}
```

What does the above program print? (GATE CS 2005)
(a) 8, 4, 0, 2, 14
(b) 8, 4, 0, 2, 0
(C) 2, 0, 4, 8, 14
(d) 2, 0, 4, 8, 0

Answer (d)
sum has no use in foo(), it is there just to confuse. Function foo() just prints all digits of a number. In main, there is one more printf statement after foo(), so one more 0 is printed after all digits of n.

Please write comments if you find any of the above answers/explanations incorrect or you want to add some more information about questions.

14. Data Structures and Algorithms | Set 5

Following questions have been asked in GATE CS exam.

1. Consider the following C function.

```
float f(float x, int y)
{
  float p, s; int i;
  for (s=1, p=1, i=1; i < y; i ++)
  {
    p*= x/i;
    s+=p;
  }
  return s;
}
```

For large values of y, the return value of the function f best approximates (GATE CS 2003)

a) x^y

b) e^x

c) ln(1 + x)

d) x^x

Answer (b)

The function f() is implementation of Taylor's Series to calculates e^x

```
e^x = 1 + x + x^2/2! + x^3/3! + ---
```

More is the value of y more precise value of e^x will be returned by f()

References:

http://en.wikipedia.org/wiki/E_%28mathematical_constant%29

2. In the worst case, the number of comparisons needed to search a singly linked list of length n for a given element is (GATE CS 2002)

a) log 2 n

b) n/2

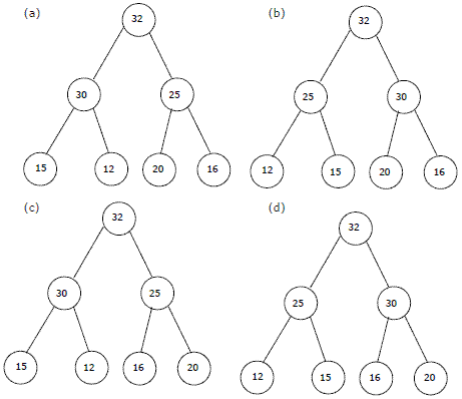
c) log 2 n – 1

d) n

Answer(d)

In the worst case, the element to be searched has to be compared with all elements of linked list.

3. The elements 32, 15, 20, 30, 12, 25, 16 are inserted one by one in the given order into a Max Heap. The resultant Max Heap is.



Answer (a)

4. Consider the following three claims
I (n + k)^m = θ(n^m), where k and m are constants
II 2^(n + 1) = 0(2^n)
III 2^(2n + 1) = 0(2^n)
Which of these claims are correct? (GATE CS 2003)
(a) I and II
(b) I and III
(c) II and III
(d) I, II and III

Answer(a)

```
(I)   (n+m)^k = n^k + c1*n^(k-1) + ... k^m = (n^k)
(II)   2^(n+1) = 2*2^n = O(2^n)
```

5. A single array A[1..MAXSIZE] is used to implement two stacks. The two stacks grow from opposite ends of the array. Variables top1 and top2 (topl< top 2) point to the location of the topmost element in each of the stacks. If the space is to be used efficiently, the condition for "stack full" is (GATE CS 2004)
a) (top1 = MAXSIZE/2) and (top2 = MAXSIZE/2+1)
b) top1 + top2 = MAXSIZE
c) (top1= MAXSIZE/2) or (top2 = MAXSIZE)
d) top1= top2 -1

Answer(d)
If we are to use space efficiently then size of the any stack can be more than MAXSIZE/2.
Both stacks will grow from both ends and if any of the stack top reaches near to the other top then stacks are full. So the condition will be top1 = top2 -1 (given that top1 < top2)

Please write comments if you find any of the above answers/explanations incorrect.

15. Data Structures and Algorithms | Set 6

Following questions have been asked in GATE CS exam.

1. The usual θ(n^2) implementation of Insertion Sort to sort an array uses linear search to identify the position where an element is to be inserted into the already sorted part of the array. If, instead, we use binary search to identify the position, the worst case running time will (GATE CS 2003)
(a) remain θ(n^2)
(b) become θ(n(log n)^2)
(c) become θ(n log n)
(d) become θ(n)

Answer (a)
If we use binary search then there will be $\lceil \log_2(n!) \rceil$ comparisons in the worst case, which is Θ(n log n) (If you want to know how $\lceil \log_2(n!) \rceil$ can be equal to Θ(n log n), then see this for proof). But the algorithm as a whole will still have a running time of Θ(n^2) on average because of the series of swaps required for each insertion.

Reference:
http://en.wikipedia.org/wiki/Insertion_sort

2. The tightest lower bound on the number of comparisons, in the worst case, for comparison-based sorting is of the order of
a) n
b) n^2
c) nlogn
d) n(log^2)n

Answer (c)
The number of comparisons that a comparison sort algorithm requires increases in proportion to nlog(n), where n is the number of elements to sort. This bound is asymptotically tight:

Given a list of distinct numbers (we can assume this because this is a worst-case analysis), there are n factorial permutations exactly one of which is the list in sorted order. The sort algorithm must gain enough information from the comparisons to identify the correct permutations. If the algorithm always completes after at most f(n) steps, it cannot distinguish more than 2^f(n) cases because the keys are distinct and each

comparison has only two possible outcomes. Therefore,

2^f(n) ≥ n!, or equivalently f(n)≥[Tex]\log_2[/Tex](n!).

References:
http://en.wikipedia.org/wiki/Comparison_sort
http://www.cs.cmu.edu/afs/cs.cmu.edu/academic/class/15451-s07/www/lecture_notes/lect0130.pdf

3. The problem 3-SAT and 2-SAT are
a) both in P
b) both NP complete
c) NP-complete and in P respectively
d) undecidable and NP-complete respectively

Answer (c)
The Boolean satisfiability problem (SAT) is a decision problem, whose instance is a Boolean expression written using only AND, OR, NOT, variables, and parentheses. The problem is: given the expression, is there some assignment of TRUE and FALSE values to the variables that will make the entire expression true? A formula of propositional logic is said to be satisfiable if logical values can be assigned to its variables in a way that makes the formula true.

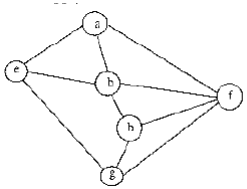
3-SAT and 2-SAT are special cases of k-satisfiability (k-SAT) or simply satisfiability (SAT), when each clause contains exactly k = 3 and k = 2 literals respectively.

2-SAT is P while 3-SAT is NP Complete. (See this for explanation)

References:

http://en.wikipedia.org/wiki/Boolean_satisfiability_problem

4. Consider the following graph



Among the following sequences
I) a b e g h f
II) a b f e h g
III) a b f h g e
IV) a f g h b e

Which are depth first traversals of the above graph? (GATE CS 2003)
a) I, II and IV only
b) I and IV only
c) II, III and IV only
d) I, III and IV only

Answer (d)

Please write comments if you find any of the above answers/explanations incorrect.

16. Principle of programming languages | Set 1

Following questions have been asked in GATE CS exam.

1.The most appropriate matching for the following pairs

```
X: Indirect addressing                  1: Loops
Y: Immediate addressing                 2: Pointers
Z: Auto decrement addressing            3. Constants
```

is (GATE CS 2000)
(a) X—3 Y—2 Z—1
(b) X—1 Y—3 Z—2
(c) X—2 Y—3 Z—1
(d) X—3 Y—1 Z—2

Answer (c)

Reference:
http://en.wikipedia.org/wiki/Addressing_mode
http://www.cs.nmsu.edu/~pfeiffer/classes/273/notes/immdirext.html

2. Aliasing in the context of programming languages refers to (GATE CS 2000)
(a) multiple variables having the same memory location
(b) multiple variables having the same value
(c) multiple variables having the same identifier
(d) multiple uses of the same variable

Answer (a)
Aliasing describes a situation in which a data location in memory can be accessed through different symbolic names in the program.

Reference

http://en.wikipedia.org/wiki/Aliasing_%28computing%29

3. What is printed by the print statements in the program P1 assuming call by reference parameter passing? (GATE CS 2001)

```
Program P1()
{
  x=10;
  y=3;
  func1(y, x, x);
  print x;
  print y;
}
func1 (x, y, z)
{
   y = y + 4;
   z = x + y + z;
}
```

a) 10, 3
b) 31, 3
c) 27, 7
d) None of the above

Answer (b)
Note the order in which parameters are passed. Inside func1(), x will actually refer to y of main(); and y and z will refer to x of main(). The Statement y = y + 4; will result in 14 and statement z = x + y + z will make z = 3 + 14 + 14 = 31 (because y and z point to same variable x of main). Since z refers to x of main(), main will print 31.

4. Consider the following program

```
Program P2
    var n: int:
     procedure W(var x: int)
     begin
         x=x+1;
         print x;
     end

     procedure D
     begin
         var n: int;
         n=3;
         W(n);
```

```
    end
begin //beginP2
  n=10;
  D;
end
```

If the language has dynamic scoping and parameters are passed by reference, what will be printed by the program? (GATE CS 2001)
a) 10
b) 11
c) 3
d) None of the above

Answer(d)
Program will print 4.

5. The- results returned by functions under value-result and reference parameter passing conventions (GATE CS 2002)

a) Do not differ
b) Differ in the presence of loops
c) Differ in all cases
d) May differ in the presence of exceptions

Answer(d)
In call-by-reference evaluation, a function receives an implicit reference to the argument, rather than a copy of its value. This typically means that the function can modify the argument- something that will be seen by its caller. Note that C doesn't support call by reference but call-by-reference can be implemented using pointers.

Call-by-value-result uses a combination of call-by-value and call-by-reference. Call-by-value-result works by creating local versions of the parameters passed in. However, the values in these local versions are copied back to the original arguments after the end of the procedure.

In case of exceptions, results may differ. Let us see below example in an arbitrary language

```
int addTwo(a, b)
{
   a = a + b;
   b = a + b;
   return b;
}
```

If call-by-value-result is used then calling addTwo(x, x) will return 3x (see below for explanation).

```
    a = a + b; will result in a = x + x
    b = a + b; will result in  b = 2x + x
```

If call-by-reference is used then addTwo(x, x) will return 4x (see below for explanation).

```
    a = a + b; will result in a = x + x
    b = a + b; will result in  b = 2x + 2x
```

References:
http://c2.com/cgi/wiki?CallByValueResult
http://en.wikipedia.org/wiki/Evaluation_strategy

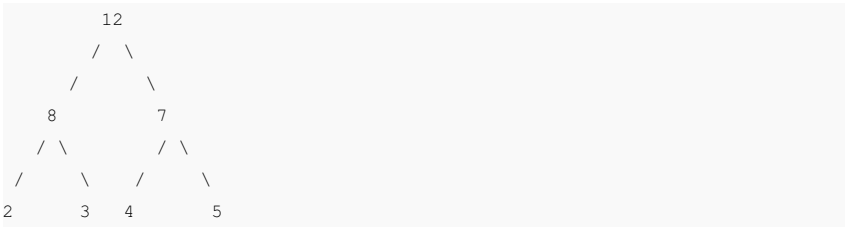
Please write comments if you find any of the above answers/explanations incorrect or you want to share more information about the topics discussed above.

17. Data Structures and Algorithms | Set 7

Following questions have been asked in GATE CS 2006 exam.

1. In a binary max heap containing n numbers, the smallest element can be found in time (GATE CS 2006)
(A) 0(n)
(B) O(logn)
(C) 0(loglogn)
(D) 0(1)

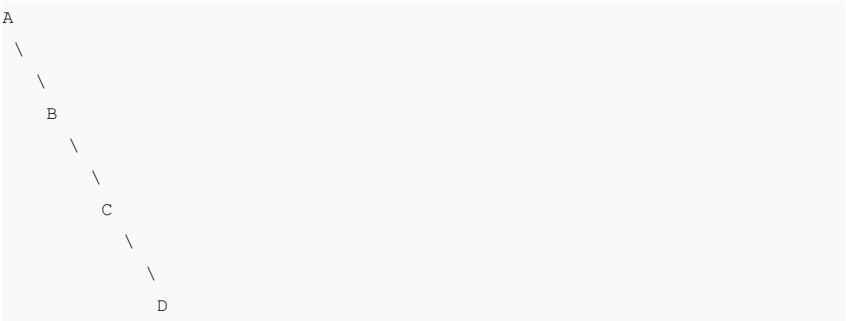
Answer (A)
In a max heap, the smallest element is always present at a leaf node. So we need to check for all leaf nodes for the minimum value. Worst case complexity will be O(n)



2. A scheme for storing binary trees in an array X is as follows. Indexing of X starts at 1 instead of 0. the root is stored at X[1]. For a node stored at X[i], the left child, if any, is stored in X[2i] and the right child, if any, in X[2i+1]. To be able to store any binary tree on n vertices the minimum size of X should be. (GATE CS 2006)

(A) log2n
(B) n
(C) 2n + 1
(D) 2^n — 1

Answer (D)
For a right skewed binary tree, number of nodes will be 2^n – 1. For example, in below binary tree, node ‘A’ will be stored at index 1, ‘B’ at index 3, ‘C’ at index 7 and ‘D’ at index 15.



3. Which one of the following in place sorting algorithms needs the minimum number of swaps? (GATE CS 2006)
(A) Quick sort
(B) Insertion sort
(C) Selection sort
(D) Heap sort

Answer (C)
For selection sort, number of swaps required is minimum (θ(n)).

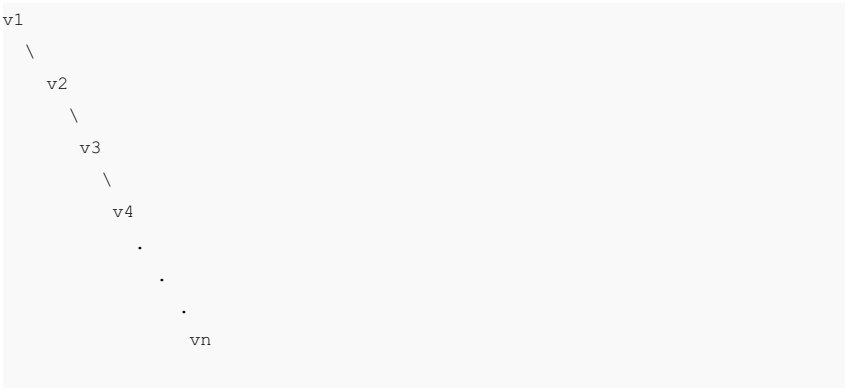
4. An element in an array X is called a leader if it is greater than all elements to the right of it in X. The best algorithm to find all leaders in an array (GATE CS 2006)
(A) Solves it in linear time using a left to right pass of the array
(B) Solves it in linear time using a right to left pass of the array
(C) Solves it using divide and conquer in time 8(nlogn)
(D) Solves it in time 8(n2)

Answer (B)
Please see this post for explanation.

5. Consider a weighted complete graph G on the vertex set {v1, v2, ..vn} such that the weight of the edge (vi, vj) is 2|i-j|. The weight of a minimum spanning tree of G is: (GATE CS 2006)
(A) n — 1
(B) 2n — 2
(C) nC2

(D) 2

Answer (B)
Minimum spanning tree of such a graph is



Weight of the minimum spanning tree
= 2|2 – 1| + 2|3 – 2| + 2|4 – 3| + 2|5 – 4| …. + 2| n – (n-1) |
= 2n – 2

Please write comments if you find any of the answers/explanations incorrect or you want to share more information about the topics discussed above.

18. Compiler Theory | Set 2

Following questions have been asked in GATE CS exam.

1. Given the following expression grammar:
E -> E * F | F+E | F
F -> F-F | id
which of the following is true? (GATE CS 2000)
(a) * has higher precedence than +
(b) – has higher precedence than *
(c) + and — have same precedence
(d) + has higher precedence than *

Answer(b)

Precedence in a grammar is enforced by making sure that a production rule with higher precedence operator will never produce an expression with operator with lower

precedence.
In the given grammar '-' has higher precedence than '*'

2. Consider a program P that consists of two source modules M1 and M2 contained in two different files. If M1 contains a reference to a function defined in M2 the reference will be resolved at (GATE CS 2004)
a) Edit time
b) Compile time
c) Link time
d) Load time

Answer (c)
Compiler transforms source code into the target language. The target language is generally in binary form known as object code. Typically, an object file can contain three kinds of symbols:

* defined symbols, which allow it to be called by other modules,
* undefined symbols, which call the other modules where these symbols are defined, and
* local symbols, used internally within the object file to facilitate relocation.

When a program comprises multiple object files, the linker combines these files into a unified executable program, resolving the symbols as it goes along.
http://en.wikipedia.org/wiki/Compiler
http://en.wikipedia.org/wiki/Linker_%28computing%29

3. Which of the following suffices to convert an arbitrary CFG to an LL(1) grammar? (GATE CS 2003)
(a) Removing left recursion alone
(b) Factoring the grammar alone
(c) Removing left recursion and factoring the grammar
(d) None of the above

Answer(d)
Removing left recursion and factoring the grammar do not suffice to convert an arbitrary CFG to LL(1) grammar.
http://pages.cpsc.ucalgary.ca/~robin/class/411/LL1.3.html

4. Assume that the SLR parser for a grammar G has n1 states and the LALR parser for G has n2 states. The relationship between nl and n2 is (GATE CS 2003)
(a) n1 is necessarily less than n2
(b) n1 is necessarily equal to n2
(c) n1 is necessarily greater than n2
(d) none of the above

Answer (b)

http://parasol.tamu.edu/people/rwerger/Courses/434/lec10.pdf
http://dragonbook.stanford.edu/lecture-notes/Stanford-CS143/11-LALR-Parsing.pdf

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

1.

19. Data Structures and Algorithms | Set 8

Following questions have been asked in GATE CS exam.

1. Consider the following functions

$f(n) = 3n^{\sqrt{n}}$

$g(n) = 2^{\sqrt{n}\log_2 n}$

$h(n) = n!$

Which of the following is true? (GATE CS 2000)
(a) h(n) is 0(f(n))
(b) h(n) is 0(g(n))
(c) g(n) is not 0(f(n))
(d) f(n) is 0(g(n))

Answer (d)
g(n) = 2^$(\sqrt{n}\log n)$ = n^$(\sqrt{n})$

f(n) and g(n) are of same asymptotic order and following statements are true.
f(n) = O(g(n))
g(n) = O(f(n)).

(a) and (b) are false because n! is of asymptotically higher order than n^$(\sqrt{n})$.

2. Let G be an undirected connected graph with distinct edge weight. Let emax be the edge with maximum weight and emin the edge with minimum weight. Which of the following statements is false? (GATE CS 2000)
(a) Every minimum spanning tree of G must contain emin
(b) If emax is in a minimum spanning tree, then its removal must disconnect G
(c) No minimum spanning tree contains emax
(d) G has a unique minimum spanning tree

Answer (c)
(a) and (b) are always true.
(c) is false because (b) is true.

(d) is true because all edge weights are distinct for G.

3. Let G be an undirected graph. Consider a depth-first traversal of G, and let T be the resulting depth-first search tree. Let u be a vertex in G and let v be the first new (unvisited) vertex visited after visiting u in the traversal. Which of the following statements is always true? (GATE CS 2000)
(a) {u,v} must be an edge in G, and u is a descendant of v in T
(b) {u,v} must be an edge in G, and v is a descendant of u in T
(c) If {u,v} is not an edge in G then u is a leaf in T
(d) If {u,v} is not an edge in G then u and v must have the same parent in T

Answer (c)

4. Consider an undirected unweighted graph G. Let a breadth-first traversal of G be done starting from a node r. Let d(r, u) and d(r, v) be the lengths of the shortest paths from r to u and v respectively, in G. If u is visited before v during the breadth-first traversal, which of the following statements is correct? (GATE CS 2001)
a) d(r, u) < d (r, v)
b) d(r, u) > d(r, v)
c) d(r, u) <= d (r, v)
d) None of the above

Answer (c)
d(r, u) and d(r, v) will be equal when u and v are at same level, otherwise d(r, u) will be less than d(r, v)

5. How many undirected graphs (not necessarily connected) can be constructed out of a given set V= {V 1, V 2,…V n} of n vertices ? (GATE CS 2001)
a) n(n-l)/2
b) 2^n
c) n!
d) 2^(n(n-1)/2)

Answer (d)
In an undirected graph, there can be maximum n(n-1)/2 edges. We can choose to have (or not have) any of the n(n-1)/2 edges. So, total number of undirected graphs with n vertices is 2^(n(n-1)/2).

Please write comments if you find any of the answers/explanations incorrect or you want to share more information about the topics discussed above.

20. Operating Systems | Set 4

Following questions have been asked in GATE CS exam.

1. Using a larger block size in a fixed block size file system leads to (GATE CS 2003)
a) better disk throughput but poorer disk space utilization
b) better disk throughput and better disk space utilization
c) poorer disk throughput but better disk space utilization
d) poorer disk throughput and poorer disk space utilization

Answer (a)
If block size is large then seek time is less (fewer blocks to seek) and disk performance is improved, but remember larger block size also causes waste of disk space.

2. Consider the following statements with respect to user-level threads and kernel supported threads
i. context switch is faster with kernel-supported threads
ii. for user-level threads, a system call can block the entire process
iii. Kernel supported threads can be scheduled independently
iv. User level threads are transparent to the kernel

Which of the above statements are true? (GATE CS 2004)
a) (ii), (iii) and (iv) only
b) (ii) and (iii) only
c) (i) and (iii) only
d) (i) and (ii) only

Answer(a)

http://en.wikipedia.org/wiki/Thread_%28computer_science%29

3. The minimum number of page frames that must be allocated to a running process in a virtual memory environment is determined by (GATE CS 2004)
a) the instruction set architecture
b) page size
c) physical memory size
d) number of processes in memory

Answer (a)
Each process needs minimum number of pages based on instruction set architecture.
Example IBM 370: 6 pages to handle MVC (storage to storage move) instruction
Instruction is 6 bytes, might span 2 pages.
2 pages to handle from.

2 pages to handle to.

4. In a system with 32 bit virtual addresses and 1 KB page size, use of one-level page tables for virtual to physical address translation is not practical because of (GATE CS 2003)
a) the large amount of internal fragmentation
b) the large amount of external fragmentation
c) the large memory overhead in maintaining page tables
d) the large computation overhead in the translation process

Answer (c)
Since page size is too small it will make size of page tables huge.

```
Size of page table =
  (total number of page table entries) *(size of a page table entry)
```

Let us see how many entries are there in page table

```
Number of entries in page table =
                    (virtual address space size)/(page size)
                    = (2^32)/(2^10)
                    = 2^22
```

Now, let us see how big each entry is.

If size of physical memory is 512 MB then number of bits required to address a byte in 512 MB is 29. So, there will be (512MB)/(1KB) = (2^29)/(2^10) page frames in physical memory. To address a page frame 19 bits are required. Therefore, each entry in page table is required to have 19 bits.

```
Note that page table entry also holds auxiliary information about the page such
as a present bit, a dirty or modified bit, address space or process ID information,
amongst others. So size of page table
    > (total number of page table entries) *(size of a page table entry)
    > (2^22 *19) bytes
    > 9.5 MB
```

And this much memory is required for each process because each process maintains its own page table. Also, size of page table will be more for physical memory more than 512MB. Therefore, it is advised to use multilevel page table for such scenarios.

References:
http://barbara.stattenfield.org/ta/cs162/section-notes/sec8.txt
http://en.wikipedia.org/wiki/Page_table

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

21. C Language | Set 4

Following questions have been asked in GATE CS exam.

1. In the C language (GATE CS 2002)
a) At most one activation record exists between the current activation record and the activation record for the main
b) The number of activation records between the current activation record and the activation record for the main depends on the actual function calling sequence.
c) The visibility of global variables depends on the actual function calling sequence.
d) Recursion requires the activation record for the recursive function to be saved on a different stack before the recursive function can be called.

Answer(b)
a) –> There is no such restriction in C language
b) –> True
c) –> False. In C, variables are statically scoped, not dynamically.
c) –> False. The activation records are stored on the same stack.

2. Consider the C program shown below.

```
# include <stdio.h>
# define print(x)  printf ("%d", x)
int x;
void Q(int z)
{
  z += x;
  print(z);
}
void P(int *y)
{
  int x = *y+2;
  Q(x);
  *y = x-1;
  print(x);
}

main(void)
{
  x=5;
  P(&x);
  print(x);
  getchar();
}
```

The output of this program is (GATE CS 2003)
a) 1276
b) 22 12 11

c) 14 6 6
d) 766

Answer (a)

Note that main() and Q() are accessing the global variable x. Inside P(), pointer variable y also holds address of global variable x, but x in P() is its P's own local variable.

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

22. Data Structures and Algorithms | Set 9

Follow questions have been asked in GATE CS exam.

1 In a heap with n elements with the smallest element at the root, the 7th smallest element can be found in time (GATE CS 2003)
a) θ(n log n)
b) θ(n)
c) θ(log n)
d) θ(1)

Answer(d)
The 7th smallest element must be in first 7 levels. Total number of nodes in any Binary Heap in first 7 levels is at most 1 + 2 + 4 + 8 + 16 + 32 + 64 which is a constant. Therefore we can always find 7th smallest element in θ(1) time.

2. Suppose the numbers 7, 5, 1, 8, 3, 6, 0, 9, 4, 2 are inserted in that order into an initially empty binary search tree. The binary search tree uses the usual ordering on natural numbers. What is the in-order traversal sequence of the resultant tree? (GATE CS 2003)
a) 7 5 1 0 3 2 4 6 8 9
b) 0 2 4 3 1 6 5 9 8 7
c) 0 1 2 3 4 5 6 7 8 9
d) 9 8 6 4 2 3 0 1 5 7

Answer (c)
In-order traversal of a BST gives elements in increasing order. So answer c is correct without any doubt.

3. Let S be a stack of size n >= 1. Starting with the empty stack, suppose we push the first n natural numbers in sequence, and then perform n pop operations. Assume that Push and Pop operation take X seconds each, and Y seconds elapse between the end of one such stack operation and the start of the next operation. For m >= 1, define the stack-life of m as the time elapsed from the end of Push(m) to the start of the pop operation that removes m from S. The average stack-life of an element of this stack is (GATE CS 2003)
a) n(X+ Y)
b) 3Y + 2X
c) n(X + Y)-X
d) Y + 2X

Answer(c)
We can easily arrive at the result by taking few examples.

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

23. Data Structures and Algorithms | Set 10

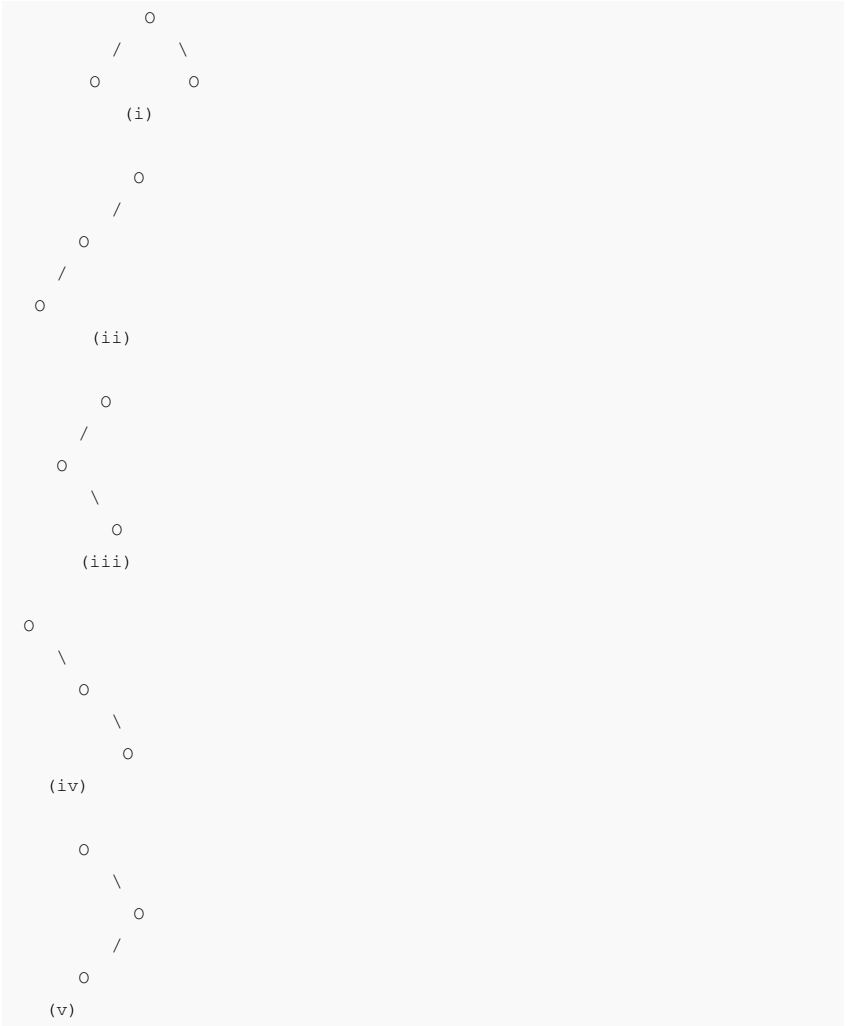
Following questions have been asked in GATE CS 2007 exam.

1. The height of a binary tree is the maximum number of edges in any root to leaf path. The maximum number of nodes in a binary tree of height h is:
(A) 2^h -1
(B) 2^(h-1) – 1
(C) 2^(h+1) -1
(D) 2*(h+1)

Answer (C)
Maximum number of nodes will be there for a complete tree.
Number of nodes in a complete tree of height h = 1 + 2 + 2^2 + 2*3 + …. 2^h = 2^(h+1) – 1

2: The maximum number of binary trees that can be formed with three unlabeled nodes is:
(A) 1
(B) 5
(C) 4
(D) 3

Answer (B)



Note that nodes are unlabeled. If the nodes are labeled, we get more number of trees.

3. Which of the following sorting algorithms has the lowest worst-case complexity?
(A) Merge sort
(B) Bubble sort
(C) Quick sort
(D) Selection sort

Answer (A)

Worst case complexities for the above sorting algorithms are as follows:
Merge Sort — nLogn

Bubble Sort — n^2
Quick Sort — n^2
Selection Sort — n^2

4. The following postfix expression with single digit operands is evaluated using a stack:

```
8 2 3 ^ / 2 3 * + 5 1 * -
```

Note that ^ is the exponentiation operator. The top two elements of the stack after the first * is evaluated are:

(A) 6, 1
(B) 5, 7
(C) 3, 2
(D) 1, 5

Answer (A)
The algorithm for evaluating any postfix expression is fairly straightforward:

```
1. While there are input tokens left
    o Read the next token from input.
    o If the token is a value
       + Push it onto the stack.
    o Otherwise, the token is an operator
      (operator here includes both operators, and functions).
       * It is known a priori that the operator takes n arguments.
       * If there are fewer than n values on the stack
         (Error) The user has not input sufficient values in the expression.
       * Else, Pop the top n values from the stack.
       * Evaluate the operator, with the values as arguments.
       * Push the returned results, if any, back onto the stack.
2. If there is only one value in the stack
    o That value is the result of the calculation.
3. If there are more values in the stack
    o (Error)  The user input has too many values.
```

Source for algorithm:
http://en.wikipedia.org/wiki/Reverse_Polish_notation#The_postfix_algorithm

Let us run the above algorithm for the given expression.
First three tokens are values, so they are simply pushed. After pushing 8, 2 and 3, the stack is as follows

```
    8, 2, 3
```

When ^ is read, top two are popped and power(2^3) is calculated

```
    8, 8
```

When / is read, top two are popped and division(8/8) is performed

```
    1
```

Next two tokens are values, so they are simply pushed. After pushing 2 and 3, the stack is as follows

```
    1, 2, 3
```

When * comes, top two are popped and multiplication is performed.

```
    1, 6
```

5. The inorder and preorder traversal of a binary tree are d b e a f c g and a b d e c f g, respectively. The postorder traversal of the binary tree is:
(A) d e b f g c a
(B) e d b g f c a
(C) e d b f g c a
(D) d e f g b c a

Answer (A)

```
Below is the given tree.
                              a
                           /    \
                         /        \
                       b            c
                     /   \        /   \
                   /       \    /       \
                 d         e    f         g
```

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

24. Data Structures and Algorithms | Set 11

Following questions have been asked in GATE CS 2007 exam.

1. Consider a hash table of size seven, with starting index zero, and a hash

function (3x + 4)mod7. Assuming the hash table is initially empty, which of the following is the contents of the table when the sequence 1, 3, 8, 10 is inserted into the table using closed hashing? Note that '_' denotes an empty location in the table.

(A) 8, _, _, _, _, _, 10
(B) 1, 8, 10, _, _, _, 3
(C) 1, _, _, _, _, _,3
(D) 1, 10, 8, _, _, _, 3

Answer (B)
Please see http://lcm.csa.iisc.ernet.in/dsa/node38.html for closed hashing and probing.

Let us put values 1, 3, 8, 10 in the hash of size 7.

Initially, hash table is empty

```
-    -    -    -    -    -    -
0    1    2    3    4    5    6
```

The value of function (3x + 4)mod 7 for 1 is 0, so let us put the value at 0

```
1    -    -    -    -    -    -
0    1    2    3    4    5    6
```

The value of function (3x + 4)mod 7 for 3 is 6, so let us put the value at 6

```
1    -    -    -    -    -    3
0    1    2    3    4    5    6
```

The value of function (3x + 4)mod 7 for 8 is 0, but 0 is already occupied, let us put the value(8) at next available space(1)

```
1    8    -    -    -    -    3
0    1    2    3    4    5    6
```

The value of function (3x + 4)mod 7 for 10 is 6, but 6 is already occupied, let us put the value(10) at next available space(2)

```
1    8   10    -    -    -    3
0    1    2    3    4    5    6
```

2. In an unweighted, undirected connected graph, the shortest path from a node S to every other node is computed most efficiently, in terms of time complexity by

(A) Dijkstra's algorithm starting from S.
(B) Warshall's algorithm
(C) Performing a DFS starting from S.
(D) Performing a BFS starting from S.

Answer(D)

```
* Time Comlexity of the Dijkstra's algorithm is O(|V|^2 + E)
* Time Comlexity of the Warshall's algorithm is O(|V|^3)
* DFS cannot be used for finding shortest paths
* BFS can be used for unweighted graphs. Time Complexity for BFS is O(|E| + |V|)
```

3. A complete n-ary tree is a tree in which each node has n children or no children. Let I be the number of internal nodes and L be the number of leaves in a complete n-ary tree. If L = 41, and I = 10, what is the value of n?
(A) 3
(B) 4
(C) 5
(D) 6

Answer (C)
For an n-ary tree where each node has n children or no children, following relation holds

```
    L = (n-1)*I + 1
```

Where L is the number of leaf nodes and I is the number of internal nodes.

Let us find out the value of n for the given data.

```
L = 41 , I = 10
41 = 10*(n-1) + 1
(n-1) = 4
n = 5
```

4. In the following C function, let n >= m.

```
int gcd(n,m)
{
  if (n%m ==0) return m;
  n = n%m;
  return gcd(m,n);
}
```

How many recursive calls are made by this function?
(A) θ(logn)?
(B) Ω(n)
(C) θ(loglogn)
(D) θ(sqrt(n))

Answer (A)
Above code is implementation of the Euclidean algorithm for finding Greatest Common Divisor (GCD).
Please see http://mathworld.wolfram.com/EuclideanAlgorithm.html for time complexity.

5. What is the time complexity of the following recursive function:

```
int DoSomething (int n)
{
  if (n <= 2)
    return 1;
  else
    return (DoSomething (floor(sqrt(n))) + n);
}
```

(A) θ(n)
(B) θ(nlogn)
(C) θ(logn)
(D) θ(loglogn)

Answer (D)
Recursive relation for the DoSomething() is

```
T(n) =  T() + C1 if n > 2
```

We have ignored the floor() part as it doesn't matter here if it's a floor or ceiling.

```
Let n = 2^m,  T(n) = T(2^m)
Let T(2^m) =  S(m)

From the above two, T(n) = S(m)

S(m) = S(m/2) + C1  /* This is simply binary search recursion*/
S(m)  = O(logm)
        = O(loglogn)  /* Since n = 2^m */

Now, let us go back to the original recursive function T(n)
T(n)  = S(m)
        = O(LogLogn)
```

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

25. Data Structures and Algorithms | Set 12

Following questions have been asked in GATE CS 2007 exam.

1. Consider the following C program segment where CellNode represents a node in a binary tree:

```
struct CellNode
{
  struct CellNOde *leftChild;
  int element;
  struct CellNode *rightChild;
};

int GetValue(struct CellNode *ptr)
{
  int value = 0;
  if (ptr != NULL)
  {
   if ((ptr->leftChild == NULL) &&
        (ptr->rightChild == NULL))
      value = 1;
   else
      value = value + GetValue(ptr->leftChild)
                    + GetValue(ptr->rightChild);
  }
  return(value);
}
```

The value returned by GetValue() when a pointer to the root of a binary tree is passed as its argument is:

(A) the number of nodes in the tree

(B) the number of internal nodes in the tree

(C) the number of leaf nodes in the tree

(D) the height of the tree

Answer (C)

For explanation, please see our post http://geeksforgeeks.org/?p=2755 for counting leaf nodes.

2. Consider the process of inserting an element into a Max Heap, where the Max Heap is represented by an array. Suppose we perform a binary search on the path from the new leaf to the root to find the position for the newly inserted element, the number of comparisons performed is:

(A) θ(logn)

(B) θ(LogLogn)

(C) θ(n)

(D) θ(nLogn)

Answer (B)

The height of a Max Heap is θ(logn). If we perform binary search for finding the correct position then we need to do θ(LogLogn) comparisons.

3. Let w be the minimum weight among all edge weights in an undirected connected graph. Let e be a specific edge of weight w . Which of the following is FALSE?

(A) There is a minimum spanning tree containing e.

(B) If e is not in a minimum spanning tree T, then in the cycle formed by adding e to T, all edges have the same weight.
(C) Every minimum spanning tree has an edge of weight w .
(D) e is present in every minimum spanning tree.

Answer (D)
(A), (B) and (C) are correct.
(D) is incorrect as there may be many edges of wight w in the graph and e may not be picked up in some of the minimum spanning trees.

4. An array of n numbers is given, where n is an even number. The maximum as well as the minimum of these n numbers needs to be determined. Which of the following is TRUE about the number of comparisons needed?
(A) At least 2n – c comparisons, for some constant c, are needed.
(B) At most 1.5n – 2 comparisons are needed.
(C) At least nLog2n comparisons are needed.
(D) None of the above.

Answer (B)

Please see the post http://geeksforgeeks.org/?p=4583 for details.

5. Consider the following C code segment:

```
int IsPrime(n)
{
  int i,n;
  for(i=2;i<=sqrt(n);i++)
   if(n%i == 0)
     {printf("Not Prime\n"); return 0;}
  return 1;
}
```

Let T(n) denotes the number of times the for loop is executed by the program on input n. Which of the following is TRUE?
(A) T(n) = O(sqrt(n)) and T(n) = Ω(sqrt(n))
(B) T(n) = O(sqrt(n)) and T(n) = Ω(1)
(C) T(n) = O(n) and T(n) = Ω(sqrt(n))
(D) None of the above

Answer (B)
Big O notation describes the upper bound and Big Omega notation describes the lower bound for an algorithm.

The *for* loop in the question is run maximum sqrt(n) times and minimum 1 time.
Therefore, T(n) = O(sqrt(n)) and T(n) = Ω(1)

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

26. Data Structures and Algorithms | Set 13

Following questions have been asked in GATE CS 2002 exam

1. The number of leaf nodes in a rooted tree of n nodes, with each node having 0 or 3 children is:
a) n/2
b) (n-1)/3
c) (n-1)/2
d) (2n+1)/3

Answer(d)

Let L be the number of leaf nodes and I be the number of internal nodes, then following relation holds for above given tree (For details, please see question 3 of http://geeksforgeeks.org/?p=4545)

```
L = (3-1)I + 1 = 2I + 1
```

Total number of nodes(n) is sum of leaf nodes and internal nodes

```
n = L + I
```

After solving above two, we get L = (2n+1)/3

2. The running time of the following algorithm

```
Procedure A(n)
If n <= 2 return(1) else return A();
```

is best described by
a) O(n)
b) O(log n)
c) O(1og log n)
d) O(1)

Answer(c)
For explanation, please see question 5 of http://geeksforgeeks.org/?p=4545

3. A weight-balanced tree is a binary tree in which for each node. The number of nodes in the left sub tree is at least half and at most twice the number of nodes in

the right sub tree. The maximum possible height (number of nodes on the path from the root to the farthest leaf) of such a tree on n nodes is best described by which of the following?

a) $\log_2 n$

b) $\log_{4/3} n$

c) $\log_3 n$

d) $\log_{3/2} n$

Answer(d)

Let the maximum possible height of a tree with n nodes is represented by H(n).

The maximum possible value of H(n) can be approximately written using following recursion

```
H(n) = H(2n/3) + 1
```

The solution of above recurrence is $\log_{3/2} n$. We can simply get it by drawing a recursion tree.

4. Consider the following algorithm for searching for a given number x in an unsorted – array A[1..n] having n distinct values:

```
1) Choose an i uniformly at random from 1..n;
2) If A[i] = x then Stop else Goto 1;
```

Assuming that x is present in A, what is the expected number of comparisons made by the algorithm before it terminates?

a) n

b) n-l

c) 2n

d) n/2

Answer(a)

If you remember the coin and dice questions, you can just guess the answer for the above.

Below is proof for the answer.

Let expected number of comparisons be E. Value of E is sum of following expression for all the possible cases.

```
number_of_comparisons_for_a_case * probability_for_the_case
```

Case 1

```
If A[i] is found in the first attempt
number of comparisons = 1
probability of the case  = 1/n
```

Case 2

```
If A[i] is found in the second attempt
number of comparisons = 2
probability of the case  = (n-1)/n*1/n
```

Case 3

```
If A[i] is found in the third attempt
number of comparisons = 2
probability of the case  = (n-1)/n*(n-1)/n*1/n
```

There are actually infinite such cases. So, we have following infinite series for E.

```
E  = 1/n + [(n-1)/n]*[1/n]*2 + [(n-1)/n]*[(n-1)/n]*[1/n]*3 + ….  (1)
```

After multiplying equation (1) with (n-1)/n, we get

```
E (n-1)/n = [(n-1)/n]*[1/n] + [(n-1)/n]*[(n-1)/n]*[1/n]*2 +
                                  [(n-1)/n]*[(n-1)/n]*[(n-1)/n]*[1/n]*3 …….(2)
```

Subtracting (2) from (1), we get

```
E/n = 1/n + (n-1)/n*1/n + (n-1)/n*(n-1)/n*1/n + ……
```

The expression on right side is a GP with infinite elements. Let us apply the sum formula (a/(1-r))

```
E/n = [1/n]/[1-(n-1)/n]  = 1
E = n
```

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

27. Data Structures and Algorithms | Set 14

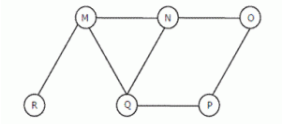
Following questions have been asked in GATE CS 2008 exam.

1. We have a binary heap on n elements and wish to insert n more elements (not necessarily one after another) into this heap. The total time required for this is
(A) θ(logn)

(B) θ(n)
(C) θ(nlogn)
(D) θ(n^2)

The worst case time complexity for insertion in a binary heap is O(Logn) (Refer Wiki).
So inserting n elements in a heap of size n should take θ(nlogn) time.
But choice **(B)** seems to be more appropriate answer. One of the solution of O(n) complexity can be to take the 'n' elements of the heap and other 'n' elements together and construct heap in O(2n) = O(n). Thanks to pankaj for suggesting this solution.

2. The Breadth First Search algorithm has been implemented using the queue data structure. One possible order of visiting the nodes of the following graph is



(A) MNOPQR
(B) NQMPOR
(C) QMNPRO
(D) QMNPOR

Answer (C)

3. Consider the following functions:

```
f(n)    = 2^n
g(n)    = n!
h(n)    = n^logn
```

Which of the following statements about the asymptotic behaviour of f(n), g(n), and h(n) is true?

(A) f(n) = O(g(n)); g(n) = O(h(n))
(B) f(n) = Ω(g(n)); g(n) = O(h(n))
(C) g(n) = O(f(n)); h(n) = O(f(n))
(D) h(n) = O(f(n)); g(n) = Ω(f(n))

Answer (D)

According to order of growth: h(n) < f(n) < g(n) (g(n) is asymptotically greater than f(n) and f(n) is asymptotically greater than h(n))
We can easily see above order by taking logs of the given 3 functions

```
lognlogn < n < log(n!)  (logs of the given f(n), g(n) and h(n)).
```

Note that log(n!) = θ(nlogn)

4. The minimum number of comparisons required to determine if an integer appears more than n/2 times in a sorted array of n integers is
(A) θ(n)
(B) θ(logn)
(C) θ(log*n)
(D) θ(n)

Answer (B)

Please see the post http://geeksforgeeks.org/?p=4722 for details.

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

28. C Language | Set 5

Following questions have been asked in GATE CS 2008 exam.

1. What is printed by the following C program?

```
int f(int x, int *py, int **ppz)
{
  int y, z;
  **ppz += 1;
   z  = **ppz;
  *py += 2;
   y = *py;
   x += 3;
   return x + y + z;
}

void main()
{
   int c, *b, **a;
   c = 4;
   b = &c;
   a = &b;
   printf( "%d", f(c,b,a));
   getchar();
}
```

(A) 18
(B) 19
(C) 21

(D) 22

Answer (B)

```
/* Explanation for the answer */

  /*below line changes value of c to 5. Note that x remains unaffected
    by this change as x is a copy of c and address of x is different f
  **ppz += 1

  /* z is changed to 5*/
  z  = **ppz;

  /* changes c to 7, x is not changed */
  *py += 2;

   /* y is changed to 7*/
  y = *py;

  /* x is incremented by 3 */
   x += 3;

  /* return 7 + 7 + 5*/
  return x + y + z;
```

2. Choose the correct option to fill ?1 and ?2 so that the program below prints an input string in reverse order. Assume that the input string is terminated by a newline character.

```
void reverse(void)
 {
  int c;
  if (?1) reverse() ;
  ?2
}
main()
{
  printf ("Enter Text ") ;
  printf ("\n") ;
  reverse();
  printf ("\n") ;
}
```

(A) ?1 is (getchar() != '\n')
?2 is getchar(c);
(B) ?1 is (c = getchar()) != '\n')
?2 is getchar(c);
(C) ?1 is (c != '\n')
?2 is putchar(c);
(D) ?1 is ((c = getchar()) != '\n')
?2 is putchar(c);

Answer(D)
getchar() is used to get the input character from the user and putchar() to print the

entered character, but before printing reverse is called again and again until '\n' is entered. When '\n' is entered the functions from the function stack run putchar() statements one by one. Therefore, last entered character is printed first.
You can try running below program

```
void reverse(void); /* function prototype */

void reverse(void)
 {
  int c;
  if (((c = getchar()) != '\n'))
    reverse();
  putchar(c);
}
main()
{
  printf ("Enter Text ") ;
  printf ("\n") ;
  reverse();
  printf ("\n") ;
  getchar();
}
```

For questions 3 & 4, consider the following C functions:

```
int f1(int n)
{
  if(n == 0 || n == 1)
    return n;
  else
    return (2*f1(n-1) + 3*f1(n-2));
}

int f2(int n)
{
  int i;
  int X[N], Y[N], Z[N] ;
  X[0] = Y[0] = Z[0] = 0;
  X[1] = 1; Y[1] = 2; Z[1] = 3;
  for(i = 2; i <= n; i++)
  {
    X[i] = Y[i-1] + Z[i-2];
    Y[i] = 2*X[i];
    Z[i] = 3*X[i];
  }
  return X[n] ;
}
```

3. The running time of f1(n) and f2(n) are

(A) θ(n) and θ(n)
(B) θ(2^n) and θ(n)
(C) θ(n) and θ(2^n)
(D) θ(2^n) and θ(2^n)

Answer (B)
For f1(), let T(n) be the function for time complexity.

```
T(n) = T(n-1) + T(n-2)
```

Above recursion is a standard one for Fibonacci Numbers. After solving the recursion, we get

```
T(n) = 1/sqrt(5)[(1 + sqrt(5))/2]^n - 1/sqrt(5)[(1 - sqrt(5))/2]^n
```

Above recursion can also be written as θ(1.618.^n)
(Please see this).

In f2(), there is a single loop, so time complexity is θ(n)

Among all the 4 given choices, (B) looks closest.

4. f1(8) and f2(8) return the values
(A) 1661 and 1640
(B) 59 and 59
(C) 1640 and 1640
(D) 1640 and 1661

Both functions perform same operation, so output is same, means either (B) or (C) is correct.
f1(2) = 2*f1(1) + 3*f1(0) = 2
f1(3) = 2*f1(2) + 3*f1(1) = 2*2 + 3*1 = 7
f1(4) = 2*f1(3) + 3*f1(2) = 2*7 + 3*2 = 20
f1(5) = 2*f1(4) + 3*f1(3) = 2*20 + 3*7 = 40 + 21 = 61

We can skip after this as the only remaining choice is (C)
f1(6) = 2*f1(5) + 3*f1(4) = 2*61 + 3*20 = 122 + 60 = 182
f1(7) = 2*f1(6) + 3*f1(5) = 2*182 + 3*61 = 364 + 183 = 547
f1(8) = 2*f1(7) + 3*f1(6) = 2*547 + 3*182 = 1094 + 546 = 1640

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

29. Data Structures and Algorithms | Set 15

Following questions have been asked in GATE CS 2008 exam.

1. The most efficient algorithm for finding the number of connected components in an undirected graph on n vertices and m edges has time complexity.
(A) θ(n)
(B) θ(m)
(C) θ(m + n)
(D) θ(mn)

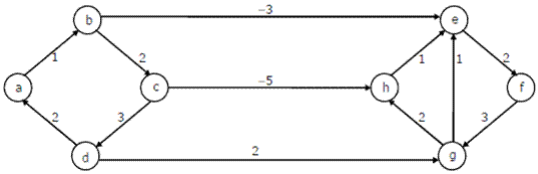
Answer (C)
Connected components can be found in O(m + n) using Tarjan's algorithm. Once we have connected components, we can count them.

2. Consider the Quicksort algorithm. Suppose there is a procedure for finding a pivot element which splits the list into two sub-lists each of which contains at least one-fifth of the elements. Let T(n) be the number of comparisons required to sort n elements. Then
(A) T(n) <= 2T(n/5) + n
(B) T(n) <= T(n/5) + T(4n/5) + n
(C) T(n) <= 2T(4n/5) + n
(D) T(n) <= 2T(n/2) + n

Answer (B)
For the case where n/5 elements are in one subset, T(n/5) comparisons are needed for the first subset with n/5 elements, T(4n/5) is for the rest 4n/5 elements, and n is for finding the pivot.

If there are more than n/5 elements in one set then other set will have less than 4n/5 elements and time complexity will be less than T(n/5) + T(4n/5) + n because recursion tree will be more balanced.

3 Dijkstra's single source shortest path algorithm when run from vertex a in the below graph, computes the correct shortest path distance to



(A) only vertex a
(B) only vertices a, e, f, g, h
(C) only vertices a, b, c, d
(D) all the vertices

Answer (D)
Dijkstra's single source shortest path is not guaranteed to work for graphs with negative weight edges, but it works for the given graph.

Let us see…

Let us run the 1st pass
b 1
b is minimum, so shortest distance to b is 1.

After 1st pass, distances are
c 3, e -2.
e is minimum, so shortest distance to e is -2

After 2nd pass, distances are
c 3, f 0.
f is minimum, so shortest distance to f is 0

After 3rd pass, distances are
c 3, g 3.
Both are same, let us take g. so shortest distance to g is 3.

After 4th pass, distances are
c 3, h 5
c is minimum, so shortest distance to c is 3

After 5th pass, distances are
h -2
h is minimum, so shortest distance to h is -2

4. The following C function takes a single-linked list of integers as a parameter and rearranges the elements of the list. The function is called with the list containing the integers 1, 2, 3, 4, 5, 6, 7 in the given order. What will be the contents of the list after the function completes execution?

```
struct node
{
  int value;
  struct node *next;
};
void rearrange(struct node *list)
{
  struct node *p, * q;
  int temp;
  if ((!list) || !list->next)
      return;
  p = list;
  q = list->next;
  while(q)
  {
     temp = p->value;
     p->value = q->value;
     q->value = temp;
     p = q->next;
     q = p?p->next:0;
  }
}
```

(A) 1,2,3,4,5,6,7
(B) 2,1,4,3,6,5,7
(C) 1,3,2,5,4,7,6
(D) 2,3,4,5,6,7,1

Answer (B)
The function rearrange() exchanges data of every node with its next node. It starts exchanging data from the first node itself.

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

30. Data Structures and Algorithms | Set 16

Following questions have been asked in GATE CS 2009 exam.

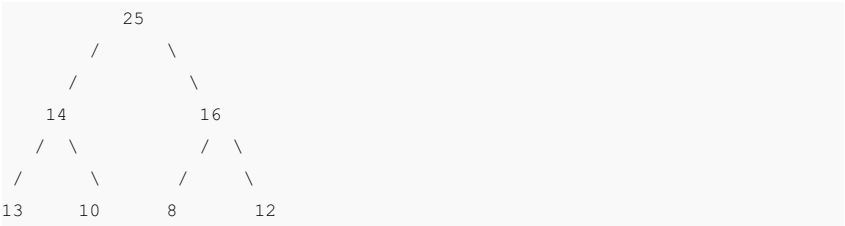
1. Consider a binary max-heap implemented using an array. Which one of the following array represents a binary max-heap?

(A) 25,12,16,13,10,8,14
(B) 25,14,13,16,10,8,12
(C) 25,14,16,13,10,8,12
(D) 25,14,12,13,10,8,16

Answer (C)

A tree is max-heap if data at every node in the tree is greater than or equal to it's children' s data.

In array representation of heap tree, a node at index i has its left child at index 2i + 1 and right child at index 2i + 2.



2. What is the content of the array after two delete operations on the correct answer to the previous question?

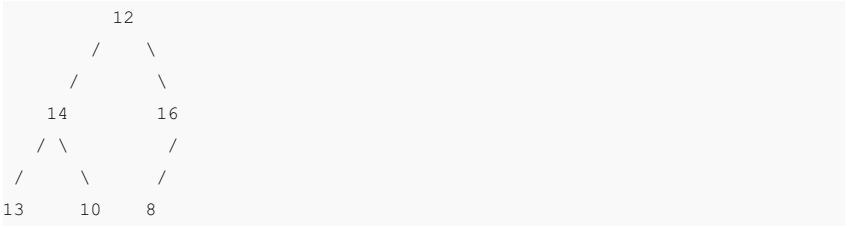
(A) 14,13,12,10,8
(B) 14,12,13,8,10
(C) 14,13,8,12,10
(D) 14,13,12,8,10

Answer(D)

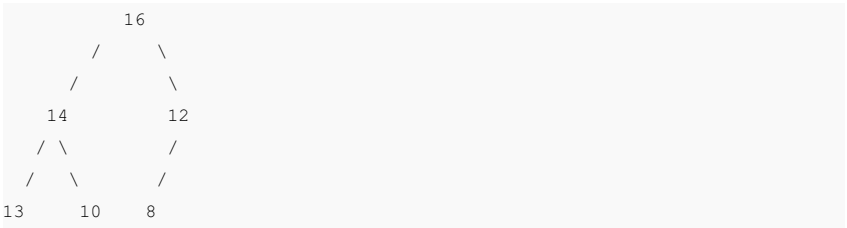
For Heap trees, deletion of a node includes following two operations.

1) Replace the root with last element on the last level.
2) Starting from root, heapify the complete tree from top to bottom..

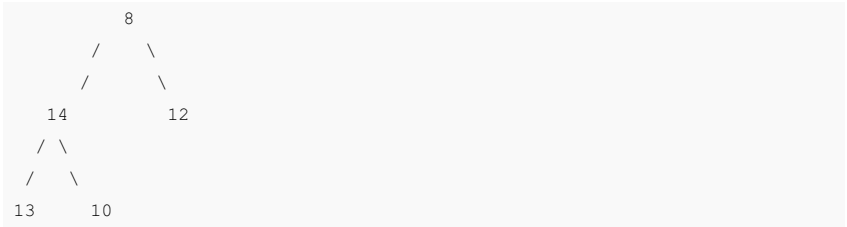
Let us delete the two nodes one by one:
1) Deletion of 25:
Replace 25 with 12



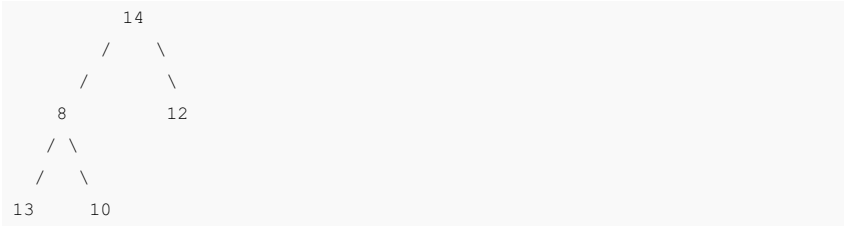
Since heap property is violated for root (16 is greater than 12), make 16 as root of the tree.

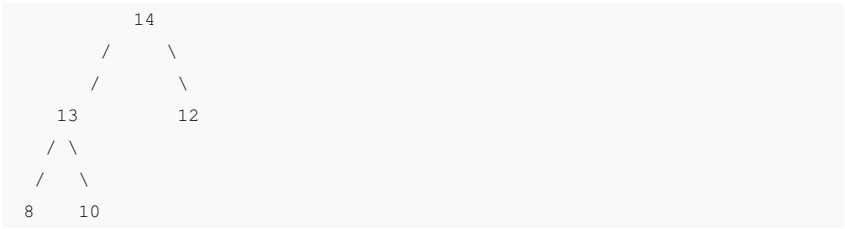


2) Deletion of 16:
Replace 16 with 8



Heapify from root to bottom.





3. In quick sort, for sorting n elements, the (n/4)th smallest element is selected as pivot using an O(n) time algorithm. What is the worst case time complexity of the quick sort?

(A) θ(n)
(B) θ(nLogn)
(C) θ(n^2)
(D) θ(n^2 log n)

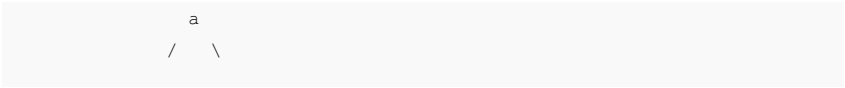
Answer(B)
The recursion expression becomes:

T(n) = T(n/4) + T(3n/4) + cn

After solving the above recursion, we get θ(nLogn).

4. What is the maximum height of any AVL-tree with 7 nodes? Assume that the height of a tree with a single node is 0.
(A) 2
(B) 3
(C) 4
(D) 5

Answer(B)
AVL trees are binary trees with the following restrictions.
1) the height difference of the children is at most 1.
2) both children are AVL trees



```
          /      \
         b        c
       /  \      /
      /    \    /
     d      e   g
    /
   /
  h
```

References:
http://en.wikipedia.org/wiki/AVL_tree

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

31. C Language | Set 6

Following questions have been asked in GATE CS 2006 exam.

1. Consider the following C-program fragment in which i, j and n are integer variables.

```
for (i = n, j = 0; i >0; i /= 2, j += i);
```

Let val(j) denote the value stored in the variable j after termination of the for loop. Which one of the following is true?
(A) val(j) = θ(logn)
(B) val(j) = θ(sqrt(n))
(C) val(j) = θ(n)
(D) val(j) = θ(nlogn)

Answer (C)

Note the semicolon after the for loop, so there is nothing in the body. The variable j is initially 0 and value of j is sum of values of i. i is initialized as n and is reduced to half in each iteration.

j = n/2 + n/4 + n/8 + .. + 1 = θ(n)

2. Consider the following C-function in which a[n] and b[m] are two sorted integer arrays and c[n + m] be another integer array.

```
void xyz(int a[], int b [], int c[])
{
  int i, j, k;
  i = j = k = 0;
  while ((i<n) && (j<m))
     if (a[i] < b[j]) c[k++] = a[i++];
     else c[k++] = b[j++];
}
```

Which of the following condition(s) hold(s) after the termination of the while loop?
(i) j < m, k = n+j-1, and a[n-1] < b[j] if i = n
(ii) i < n, k = m+i-1, and b[m-1] <= a[i] if j = m
(A) only (i)
(B) only (ii)
(C) either (i) or (ii) but not both
(D) neither (i) nor (ii)

Answer (C)
The condition (i) is true if the last inserted element in c[] is from a[] and condition (ii) is true if the last inserted element is from b[].

3. Consider this C code to swap two integers and these five statements: the code

```
void swap(int *px, int *py)
{
   *px = *px - *py;
   *py = *px + *py;
   *px = *py - *px;
}
```

S1: will generate a compilation error
S2: may generate a segmentation fault at runtime depending on the arguments passed
S3: correctly implements the swap procedure for all input pointers referring to integers stored in memory locations accessible to the process
S4: implements the swap procedure correctly for some but not all valid input pointers
S5: may add or subtract integers and pointers.
(A) S1
(B) S2 and S3
(C) S2 and S4
(D) S2 and S5

Answer (C)
S2: May generate segmentation fault if value at pointers px or py is constant or px or py points to a memory location that is invalid
S4: May not work for all inputs as arithmetic overflow can occur.

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

32. Data Structures and Algorithms | Set 17

Following questions have been asked in GATE CS 2006 exam.

1. An implementation of a queue Q, using two stacks S1 and S2, is given below:

```
void insert(Q, x) {
   push (S1, x);
}

void delete(Q){
   if(stack-empty(S2)) then
      if(stack-empty(S1)) then {
          print("Q is empty");
          return;
      }
      else while (!(stack-empty(S1))){
          x=pop(S1);
          push(S2,x);
      }
   x=pop(S2);
}
```

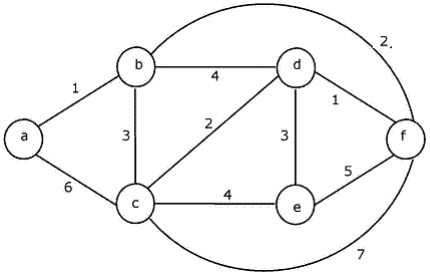
Let n insert and m (<=n) delete operations be performed in an arbitrary order on an empty queue Q. Let x and y be the number of push and pop operations performed respectively in the process. Which one of the following is true for all m and n?
(A) n+m <= x < 2n and 2m <= y <= n+m
(B) n+m <= x < 2n and 2m<= y <= 2n
(C) 2m <= x < 2n and 2m <= y <= n+m
(D) 2m <= x <2n and 2m <= y <= 2n

Answer(A)

The order in which insert and delete operations are performed matters here.
The best case: Insert and delete operations are performed alternatively. In every delete operation, 2 pop and 1 push operations are performed. So, total m+ n push (n push for insert() and m push for delete()) operations and 2m pop operations are performed.

The worst case: First n elements are inserted and then m elements are deleted. In first delete operation, n + 1 pop operations and n push operation are performed. Other than first, in all delete operations, 1 pop operation is performed. So, total m + n pop operations and 2n push operations are performed (n push for insert() and m push for delete())

2. Consider the following graph:



Which one of the following cannot be the sequence of edges added, in that order, to a minimum spanning tree using Kruskal's algorithm?
(A) (a—b),(d—f),(b—f),(d—c),(d—e)
(B) (a—b),(d—f),(d—c),(b—f),(d—e)
(C) (d—f),(a—b),(d—c),(b—f),(d—e)
(D) (d—f),(a—b),(b—f),(d—e),(d—c)

Answer (D)
The edge (d-e) cannot be considered before (d-c) in Kruskal's minimum spanning tree algorithm because Kruskal's algorithm picks the edge with minimum weight from the current set of edges at each step.

3. The median of n elements can be found in O(n)time. Which one of the following is correct about the complexity of quick sort, in which median is selected as pivot?
(A) θ(n)
(B) θ(nlogn)
(C) θ(n^2)
(D) θ(n^3)

Answer (B)
If median is always used as pivot, then recursion remains T(n) = 2T(n/2) + cn for all the cases where cn is combined time for median finding and partition. So, worst case time complexity of this quick sort becomes θ(nlogn). In practical implementations, however, this variant is considerably slower on average (see http://en.wikipedia.org/wiki/Quicksort#Selection-based_pivoting)

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

33. Data Structures and Algorithms | Set 18

Following questions have been asked in GATE CS 2006 exam.

1. Consider the polynomial p(x) = a0 + a1x + a2x^2 +a3x^3, where ai != 0, for all i.

The minimum number of multiplications needed to evaluate p on an input x is:

(A) 3
(B) 4
(C) 6
(D) 9

Answer (A)

Multiplications can be minimized using following order for evaluation of the given expression.
p(x) = a0 + x(a1 + x(a2 + a3x))

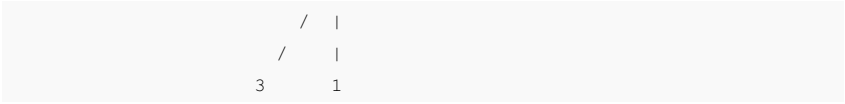
2. To implement Dijkstra's shortest path algorithm on unweighted graphs so that it runs in linear time, the data structure to be used is:
(A) Queue
(B) Stack
(C) Heap
(D) B-Tree

Answer(A)
The shortest path in an un-weighted graph means the smallest number of edges that must be traversed in order to reach the destination in the graph. This is the same problem as solving the weighted version where all the weights happen to be 1. If we use Queue (FIFO) instead of Priority Queue (Min Heap), we get the shortest path in linear time O(|V| + |E|). Basically we do BFS traversal of the graph to get the shortest paths.

3. A 3-ary max heap is like a binary max heap, but instead of 2 children, nodes have 3 children. A 3-ary heap can be represented by an array as follows: The root is stored in the first location, a[0], nodes in the next level, from left to right, is stored from a[1] to a[3]. The nodes from the second level of the tree from left to right are stored from a[4] location onward. An item x can be inserted into a 3-ary heap containing n items by placing x in the location a[n] and pushing it up the tree to satisfy the heap property.

Which one of the following is a valid sequence of elements in an array representing 3-ary max heap?
(A) 1, 3, 5, 6, 8, 9
(B) 9, 6, 3, 1, 8, 5
(C) 9, 3, 6, 8, 5, 1
(D) 9, 5, 6, 8, 3, 1

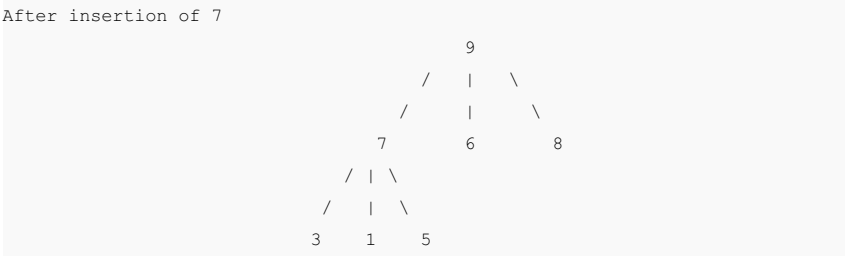
Answer (D)



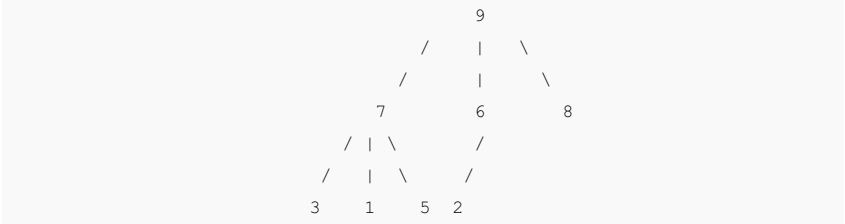
4. Suppose the elements 7, 2, 10 and 4 are inserted, in that order, into the valid 3-ary max heap found in the above question, Which one of the following is the sequence of items in the array representing the resultant heap?

(A) 10, 7, 9, 8, 3, 1, 5, 2, 6, 4
(B) 10, 9, 8, 7, 6, 5, 4, 3, 2, 1
(C) 10, 9, 4, 5, 7, 6, 8, 2, 1, 3
(D) 10, 8, 6, 9, 7, 2, 3, 4, 1, 5

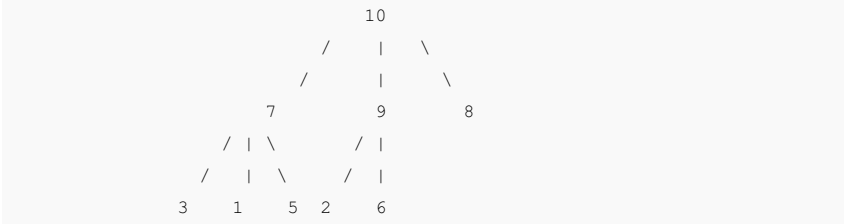
Answer(A)



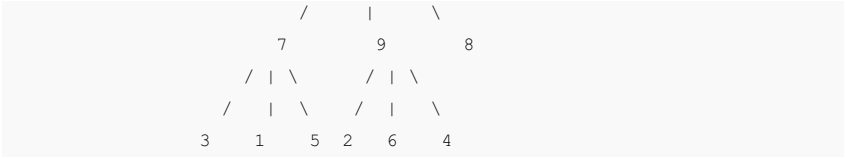
After insertion of 2



After insertion of 10



After insertion of 4



Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

34. Data Structures and Algorithms | Set 19

Following questions have been asked in GATE CS 2009 exam.

1. Let X be a problem that belongs to the class NP. Then which one of the following is TRUE?
(A) There is no polynomial time algorithm for X.
(B) If X can be solved deterministically in polynomial time, then P = NP.
(C) If X is NP-hard, then it is NP-complete.
(D) X may be undecidable.

Answer (C)
(A) is incorrect because set NP includes both P(Polynomial time solvable) and NP-Complete .
(B) is incorrect because X may belong to P (same reason as (A))
(C) is correct because NP-Complete set is intersection of NP and NP-Hard sets.
(D) is incorrect because all NP problems are decidable in finite set of operations.

2. What is the number of swaps required to sort n elements using selection sort, in the worst case?
(A) θ(n)
(B) θ(n log n)
(C) θ(n^2)
(D) θ(n^2 log n)

Answer (A)
Here is Selection Sort algorithm for sorting in ascending order.

```
1. Find the minimum value in the list
2. Swap it with the value in the first position
3. Repeat the steps above for the remainder of the list (starting at
   the second position and advancing each time)
```

As we can see from the algorithm, selection sort performs swap only after finding the

appropriate position of the current picked element. So there are O(n) swaps performed in selection sort.
Because swaps require writing to the array, selection sort is preferable if writing to memory is significantly more expensive than reading. This is generally the case if the items are huge but the keys are small. Another example where writing times are crucial is an array stored in EEPROM or Flash. There is no other algorithm with less data movement.

References:
http://en.wikipedia.org/wiki/Selection_sort

3. The running time of an algorithm is represented by the following recurrence relation:

```
    if  n <= 3  then   T(n) = n
    else T(n) = T(n/3) + cn
```

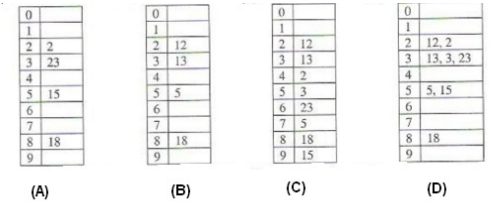
Which one of the following represents the time complexity of the algorithm?
(A) θ(n)
(B) θ(n log n)
(C) θ(n^2)
(D) θ(n^2log n)

Answer(A)

```
T(n) = cn + T(n/3)
     = cn + cn/3 + T(n/9)
     = cn + cn/3 + cn/9 + T(n/27)
Taking the sum of infinite GP series. The value of T(n) will
be less than this sum.
T(n) <= cn(1/(1-1/3))
     <= 3cn/2

or we can say
cn <= T(n) <= 3cn/2
Therefore T(n) = (n)
```

This can also be solved using Master Theorem for solving recurrences. The given expression lies in Case 3 of the theorem.

4. 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?



0	
1	
2	2
3	23
4	
5	15
6	
7	
8	18
9	

(A)

0	
1	
2	12
3	13
4	
5	5
6	
7	
8	18
9	

(B)

0	
1	
2	12
3	13
4	2
5	3
6	23
7	5
8	18
9	15

(C)

0	
1	
2	12, 2
3	13, 3, 23
4	
5	5, 15
6	
7	
8	18
9	

(D)

Answer (C)
To get the idea of open addressing concept, you can go through below lines from Wikipedia
.
Open addressing, or closed hashing, is a method of collision resolution in hash tables. With this method a hash collision is resolved by probing, or searching through alternate locations in the array (the probe sequence) until either the target record is found, or an unused array slot is found, which indicates that there is no such key in the table. Well known probe sequences include:

linear probing in which the interval between probes is fixed–often at 1.
quadratic probing in which the interval between probes increases linearly (hence, the indices are described by a quadratic function).
double hashing in which the interval between probes is fixed for each record but is computed by another hash function.

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

35. Data Structures and Algorithms | Set 20

Following questions have asked in GATE CS 2006 exam.

1. Let S be an NP-complete problem and Q and R be two other problems not known to be in NP. Q is polynomial time reducible to S and S is polynomial-time reducible to R. Which one of the following statements is true?
(A) R is NP-complete
(B) R is NP-hard
(C) Q is NP-complete
(D) Q is NP-hard

Answer (B)
(A) Incorrect because R is not in NP. A NP Complete problem has to be in both NP and NP-hard.

(B) Correct because a NP Complete problem S is polynomial time educable to R.
(C) Incorrect because Q is not in NP.
(D) Incorrect because there is no NP-complete problem that is polynomial time Turing-reducible to Q.

2) A set X can be represented by an array x[n] as follows:

$$x[i] = \begin{cases} 1 & \text{if } i \in X \\ 0 & \text{otherwise} \end{cases}$$

Consider the following algorithm in which x,y and z are Boolean arrays of size n:

```
algorithm zzz(x[] , y[], z [])
{
   int i;
   for (i=0; i<n; ++i)
     z[i] = (x[i] ^ ~y[i]) V (~x[i] ^ y[i])
}
```

The set Z computed by the algorithm is:
(A) (X Intersection Y)
(B) (X Union Y)
(C) (X-Y) Intersection (Y-X)
(D) (X-Y) Union (Y-X)

Answer (D)
The expression x[i] ^ ~y[i]) results the only 1s in x where corresponding entry in y is 0. An array with these set bits represents set X – Y
The expression ~x[i] ^ y[i]) results the only 1s in y where corresponding entry in x is 0. An array with these set bits represents set Y – X.
The operator "V" results in Union of the above two sets.

3. Consider the following recurrence:

$$T(n) = 2T\left(\lceil \sqrt{n} \rceil\right) + 1, T(1) = 1$$

Which one of the following is true?
(A) T(n) = θ(loglogn)
(B) T(n) = θ(logn)
(C) T(n) = θ(sqrt(n))
(D) T(n) = θ(n)

Answer (B)

```
Let n = 2^m
T(2^m) = T(2^(m/2)) + 1
Let T(2^m) =  S(m)
S(m)  = 2S(m/2) + 1
```

Above expression is a binary tree traversal recursion whose time complexity is θ(m).
You can also prove using Master theorem.

```
S(m)   = (m)
       = (logn)   /* Since n = 2^m */
```

Now, let us go back to the original recursive function T(n)

```
  T(n)   = T(2^m)  = S(m)
                   = (Logn)
```

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

36. Data Structures and Algorithms | Set 21

Following questions have been asked in GATE CS 2008 exam.

1. The subset-sum problem is defined as follows. Given a set of n positive integers, S = {a1 ,a2 ,a3 ,…,an} and positive integer W, is there a subset of S whose elements sum to W? A dynamic program for solving this problem uses a 2-dimensional Boolean array X, with n rows and W+1 columns. X[i, j],1 <= i <= n, 0 <= j <= W, is TRUE if and only if there is a subset of {a1 ,a2 ,...,ai} whose elements sum to j. Which of the following is valid for 2 <= i <= n and ai <= j <= W?
(A) X[i, j] = X[i – 1, j] V X[i, j -ai]
(B) X[i, j] = X[i – 1, j] V X[i – 1, j – ai]
(C) X[i, j] = X[i – 1, j] V X[i, j – ai]
(D) X[i, j] = X[i – 1, j] V X[i -1, j – ai]

Answer (B)

X[I, j] (2 <= i <= n and ai <= j <= W), is true if any of the following is true
1) Sum of weights excluding ai is equal to j, i.e., if X[i-1, j] is true.
2) Sum of weights including ai is equal to j, i.e., if X[i-1, j-ai] is true so that we get (j – ai) + ai as j.

2. In question 1, which entry of the array X, if TRUE, implies that there is a subset whose elements sum to W?
(A) X[1, W]
(B) X[n ,0]
(C) X[n, W]
(D) X[n -1, n]

Answer (C)
If we get the entry X[n, W] as true then there is a subset of {a1, a2, .. an} that has sum as

W.

Reference: http://en.wikipedia.org/wiki/Subset_sum_problem

3. Consider the following C program that attempts to locate an element x in an array Y[] using binary search. The program is erroneous.

```
1.   f(int Y[10], int x) {
2.     int i, j, k;
3.     i = 0; j = 9;
4.     do {
5.             k =  (i + j) /2;
6.             if( Y[k] < x)  i = k; else j = k;
7.         } while(Y[k] != x && i < j);
8.     if(Y[k] == x) printf ("x is in the array ") ;
9.     else printf (" x is not in the array ") ;
10. }
```

On which of the following contents of Y and x does the program fail?

(A) Y is [1 2 3 4 5 6 7 8 9 10] and x < 10

(B) Y is [1 3 5 7 9 11 13 15 17 19] and x < 1

(C) Y is [2 2 2 2 2 2 2 2 2 2] and x > 2

(D) Y is [2 4 6 8 10 12 14 16 18 20] and 2 < x < 20 and x is even

Answer (C)

The above program doesn't work for the cases where element to be searched is the last element of Y[] or greater than the last element (or maximum element) in Y[]. For such cases, program goes in an infinite loop because i is assigned value as k in all iterations, and i never becomes equal to or greater than j. So while condition never becomes false.

4. In question 3, the correction needed in the program to make it work properly is

(A) Change line 6 to: if (Y[k] < x) i = k + 1; else j = k-1;

(B) Change line 6 to: if (Y[k] < x) i = k - 1; else j = k+1;

(C) Change line 6 to: if (Y[k] <= x) i = k; else j = k;

(D) Change line 7 to: } while ((Y[k] == x) && (i < j));

Answer (A)

Below is the corrected function

```
f(int Y[10], int x) {
   int i, j, k;
   i = 0; j = 9;
   do {
           k =  (i + j) /2;
           if( Y[k] < x)  i = k + 1; else j = k - 1;
       } while(Y[k] != x && i < j);
   if(Y[k] == x) printf ("x is in the array ") ;
   else printf (" x is not in the array ") ;
}
```

Reference: http://en.wikipedia.org/wiki/Binary_search_algorithm#Implementations

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

37. Data Structures and Algorithms | Set 22

Following questions have been asked in GATE CS 2005 exam.

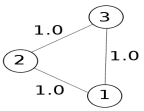
1) A program P reads in 500 integers in the range [0..100] representing the scores of 500 students. It then prints the frequency of each score above 50. What would be the best way for P to store the frequencies?
(a) An array of 50 numbers
(b) An array of 100 numbers
(c) An array of 500 numbers
(d) A dynamically allocated array of 550 numbers

Answer (a)
An array of size 50 looks the best option to store number of students for each score. We need to store frequencies of scores above 50. We can ignore scores below 50 and to index the scores above 50, we can subtract 50 from the score value/

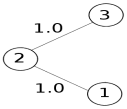
2) An undirected graph G has n nodes. Its adjacency matrix is given by an n × n square matrix whose (i) diagonal elements are 0's and (ii) non-diagonal elements are 1's. which one of the following is TRUE?
(a) Graph G has no minimum spanning tree (MST)
(b) Graph G has a unique MST of cost n-1
(c) Graph G has multiple distinct MSTs, each of cost n-1
(d) Graph G has multiple spanning trees of different costs

Answer (c)
If all non diagonal elements are 1, then every vertex is connected to every other vertex in the graph with an edge of weight 1. Such a graph has multiple distinct MSTs with cost n-1. See the below example.

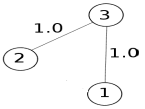
The connected graph:



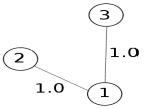
Below are three Minimum Spanning trees each of cost 2.0.
Minimum Spanning Tree 1



Minimum Spanning Tree 2



Minimum Spanning Tree 3



3) The time complexity of computing the transitive closure of a binary relation on a set of n elements is known to be:
a) O(n)
b) O(nLogn)
c) O(n^(3/2))
d) O(n^3)

Answer (d)
In mathematics, the transitive closure of a binary relation R on a set X is the smallest transitive relation on X that contains R. If the original relation is transitive, the transitive closure will be that same relation; otherwise, the transitive closure will be a different relation.

In computer science the concept of transitive closure can be thought of as constructing a data structure that makes it possible to answer reachability questions. That is, can one get from node a to node other node b in one or more hops? A binary relation tells you only that node a is connected to node b, and that node b is connected to node c, etc. After the transitive closure is constructed in an O(1) operation one may determine that node c is reachable from node a.

Warshall's algorithm can be used to construct the Transitive closure of directed graphs (). In Warshall's original formulation of the algorithm, the graph is unweighted and represented by a Boolean adjacency matrix. Then the addition operation is replaced by logical conjunction (AND) and the minimum operation by logical disjunction (OR).

References:
http://en.wikipedia.org/wiki/Floyd%E2%80%93Warshall_algorithm
http://en.wikipedia.org/wiki/Transitive_closure

4. A Priority-Queue is implemented as a Max-Heap. Initially, it has 5 elements. The level-order traversal of the heap is given below:
10, 8, 5, 3, 2

Two new elements ”1‘ and ”7‘ are inserted in the heap in that order. The level-order traversal of the heap after the insertion of the elements is:
(a) 10, 8, 7, 5, 3, 2, 1
(b) 10, 8, 7, 2, 3, 1, 5
(c) 10, 8, 7, 1, 2, 3, 5
(d) 10, 8, 7, 3, 2, 1, 5

Answer (D)

Original Max-Heap is:

```
       10
      /  \
     8    5
    / \
   3   2
```

After Insertion of 1.

```
        10
       /    \
      8      5
     / \    /
    3   2 1
```

After Insertion of 7.

```
        10
       /   \
      8     7
     / \   / \
    3   2  1   5
```

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

38. Data Structures and Algorithms | Set 23

Following questions have been asked in GATE CS 2005 exam.

1. Which one of the following is a key factor for preferring B-trees to binary search trees for indexing database relations?
(a) Database relations have a large number of records
(b) Database relations are sorted on the primary key

(c) B-trees require less memory than binary search trees
(d) Data transfer form disks is in blocks.

Answer (d)
A disk block contains fairly large number of keys. Unlike BST where each node contains only one key, B-Tree is designed to contain large number of keys so that tree height is small.

2. How many distinct binary search trees can be created out of 4 distinct keys?
(a) 5
(b) 14
(c) 24
(d) 42

Answer (b)

Here is a systematic way to enumerate these BSTs. Consider all possible binary search trees with each element at the root. If there are n nodes, then for each choice of root node, there are n – 1 non-root nodes and these non-root nodes must be partitioned into those that are less than a chosen root and those that are greater than the chosen root.

Let's say node i is chosen to be the root. Then there are i – 1 nodes smaller than i and n – i nodes bigger than i. For each of these two sets of nodes, there is a certain number of possible subtrees.

Let t(n) be the total number of BSTs with n nodes. The total number of BSTs with i at the root is t(i – 1) t(n – i). The two terms are multiplied together because the arrangements in the left and right subtrees are independent. That is, for each arrangement in the left tree and for each arrangement in the right tree, you get one BST with i at the root.

Summing over i gives the total number of binary search trees with n nodes.

$$t(n) = \sum_{i=1}^{n} t(i-1)\ t(n-i).$$

The base case is t(0) = 1 and t(1) = 1, i.e. there is one empty BST and there is one BST with one node.

$$t(2) = t(0)t(1) + t(1)t(0) = 2$$
$$t(3) = t(0)t(2) + t(1)t(1) + t(2)t(0) = 2 + 1 + 2 = 5$$
$$t(4) = t(0)t(3) + t(1)t(2) + t(2)t(1) + t(3)t(0) = 5 + 2 + 2 + 5 = 14$$

3. In a complete k-ary tree, every internal node has exactly k children. The number of leaves in such a tree with n internal nodes is:
(a) nk
(b) (n – 1) k+ 1
(c) n(k – 1) + 1

(d) n(k – 1)

Answer (c)

4) Suppose T(n) = 2T(n/2) + n, T(0) = T(1) = 1
Which one of the following is false.
a) T(n) = O(n^2)
b) T(n) = θ(nLogn)
c) T(n) = Ω(n^2)
d) T(n) = O(nLogn)

Answer (c)
The given recurrence relation can be solved using Master Theorem. It lies in case 2 of Master Theorem. Or, if you remember recurrence relation of Merge Sort or best case Quick Sort, you can guess the value of T(n).

T(n) = θ(nLogn)

By definition of Big O notation, we can say.
θ(nLogn) = O(nLogn) = O(n^2)

θ(nLogn) ca be equal to Ω(n) or Ω(nLogn), but not Ω(n^2)

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

39. C Language | Set 7

Following questions have been asked in GATE CS 2010 exam.

1. What does the following program print?

```
#include<stdio.h>
void f(int *p, int *q)
{
  p = q;
 *p = 2;
}
int i = 0, j = 1;
int main()
{
  f(&i, &j);
  printf("%d %d \n", i, j);
  getchar();
  return 0;
}
```

(A) 2 2

(B) 2 1
(C) 0 1
(D) 0 2

Answer (D)
See below f() with comments for explanation.

```
/* p points to i and q points to j */
void f(int *p, int *q)
{
  p = q;    /* p also points to j now */
 *p = 2;   /* Value of j is changed to 2 now */
}
```

2. What is the value printed by the following C program?

```
#include<stdio.h>
int f(int *a, int n)
{
  if(n <= 0) return 0;
  else if(*a % 2 == 0) return *a + f(a+1, n-1);
  else return *a - f(a+1, n-1);
}

int main()
{
  int a[] = {12, 7, 13, 4, 11, 6};
  printf("%d", f(a, 6));
  getchar();
  return 0;
}
```

(A) -9
(B) 5
(C) 15
(D) 19

Answer (C)
f() is a recursive function which adds f(a+1, n-1) to *a if *a is even. If *a is odd then f() subtracts f(a+1, n-1) from *a. See below recursion tree for execution of f(a, 6).
.

```
f(add(12), 6) /*Since 12 is first element. a contains address of 12 */
    |
    |
12 + f(add(7), 5) /* Since 7 is the next element, a+1 contains address of 7 */
        |
        |
     7 - f(add(13), 4)
              |
              |
           13 - f(add(4), 3)
```

```
                    |
                    |
              4 + f(add(11), 2)
                        |
                        |
                  11 - f(add(6), 1)
                           |
                           |
                        6 + 0
```

So, the final returned value is 12 + (7 – (13 – (4 + (11 – (6 + 0))))) = 15

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

40. Data Structures and Algorithms | Set 24

Following questions have been asked in GATE CS 2010 exam.

1. The following C function takes a simply-linked list as input argument. It modifies the list by moving the last element to the front of the list and returns the modified list. Some part of the code is left blank.

```
typedef struct node
{
  int value;
  struct node *next;
}Node;

Node *move_to_front(Node *head)
{
  Node *p, *q;
  if ((head == NULL: || (head->next == NULL))
    return head;
  q = NULL; p = head;
  while (p-> next !=NULL)
  {
    q = p;
    p = p->next;
  }
  _____________________________
  return head;
}
```

Choose the correct alternative to replace the blank line.

(A) q = NULL; p->next = head; head = p;

(B) q->next = NULL; head = p; p->next = head;

(C) head = p; p->next = q; q->next = NULL;

(D) q->next = NULL; p->next = head; head = p;

Answer(D)
When the while loop ends, q contains address of second last node and p contains address of last node. So we need to do following things after while loop.
i) Set next of q as NULL (q->next = NULL).
ii) Set next of p as head (p->next = head).
iii) Make head as p (head = p)
Step (ii) must be performed before step (iii). If we change head first, then we lose track of head node in the original linked list.
See http://geeksforgeeks.org/?p=6850 for more details.

2. A hash table of length 10 uses open addressing with hash function h(k)=k mod 10, and linear probing. After inserting 6 values into an empty hash table, the table is as shown below.

0	
1	
2	42
3	23
4	34
5	52
6	46
7	33
8	
9	

Which one of the following choices gives a possible order in which the key values could have been inserted in the table?
(A) 46, 42, 34, 52, 23, 33
(B) 34, 42, 23, 52, 33, 46
(C) 46, 34, 42, 23, 52, 33
(D) 42, 46, 33, 23, 34, 52

Answer (C)
The sequence (A) doesn't create the hash table as the element 52 appears before 23 in this sequence.
The sequence (B) doesn't create the hash table as the element 33 appears before 46 in this sequence.
The sequence (C) creates the hash table as 42, 23 and 34 appear before 52 and 33, and 46 appears before 33.
The sequence (D) doesn't create the hash table as the element 33 appears before 23 in this sequence.

3. How many different insertion sequences of the key values using the same hash function and linear probing will result in the hash table shown above?
(A) 10
(B) 20
(C) 30
(D) 40

Answer (C)
In a valid insertion sequence, the elements 42, 23 and 34 must appear before 52 and 33, and 46 must appear before 33.
Total number of different sequences = **3! x 5** = 30
In the above expression, **3!** is for elements 42, 23 and 34 as they can appear in any order, and **5** is for element 46 as it can appear at 5 different places.

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

41. Data Structures and Algorithms | Set 25

Following questions have been asked in GATE 2010 exam.

1 Consider a complete undirected graph with vertex set {0, 1, 2, 3, 4}. Entry Wij in the matrix W below is the weight of the edge {i, j}.

$$W = \begin{pmatrix} 0 & 1 & 8 & 1 & 4 \\ 1 & 0 & 12 & 4 & 9 \\ 8 & 12 & 0 & 7 & 3 \\ 1 & 4 & 7 & 0 & 2 \\ 4 & 9 & 3 & 2 & 0 \end{pmatrix}$$

What is the minimum possible weight of a spanning tree T in this graph such that vertex 0 is a leaf node in the tree T?
(A) 7
(B) 8
(C) 9
(D) 10

Answer (D)

To get the minimum spanning tree with vertex 0 as leaf, first remove 0th row and 0th column and then get the minimum spanning tree (MST) of the remaining graph. Once we have MST of the remaining graph, connect the MST to vertex 0 with the edge with minimum weight (we have two options as there are two 1s in 0th row).

2. In the graph given in question 1, what is the minimum possible weight of a path P from vertex 1 to vertex 2 in this graph such that P contains at most 3 edges?
(A) 7
(B) 8
(C) 9
(D) 10

Answer (B)

Path: 1 -> 0 -> 4 -> 2
Weight: 1 + 4 + 3

3. The degree sequence of a simple graph is the sequence of the degrees of the nodes in the graph in decreasing order. Which of the following sequences can not be the degree sequence of any graph?
I. 7, 6, 5, 4, 4, 3, 2, 1
II. 6, 6, 6, 6, 3, 3, 2, 2
III. 7, 6, 6, 4, 4, 3, 2, 2
IV. 8, 7, 7, 6, 4, 2, 1, 1

(A) I and II
(B) III and IV
(C) IV only
(D) II and IV

Answer (D)

In sequence IV, we have a vertex with degree 8 which is not possible in a simple graph (no self loops and no multiple edges) with total vertex count as 8. Maximum possible degree in such a graph is 7.

In sequence II, four vertices are connected to 6 other vertices, but remaining 4 vertices have degrees as 3, 3, 2 and 2 which are not possible in a simple graph (no self loops and no multiple edges).

4. Consider a B+-tree in which the maximum number of keys in a node is 5. What is the minimum number of keys in any non-root node?
(A) 1
(B) 2
(C) 3
(D) 4

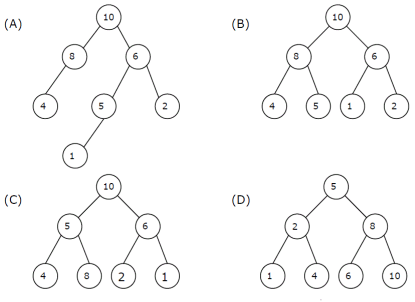
Answer (B)
Since the maximum number of keys is 5, maximum number of children a node can have is 6. By definition of B Tree, minimum children that a node can have would be 6/2 = 3. Therefore, minimum number of keys that a node can have becomes 2 (3-1).

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

42. Data Structures and Algorithms | Set 26

Following questions have been asked in GATE 2011 exam.

1) A max-heap is a heap where the value of each parent is greater than or equal to the values of its children. Which of the following is a max-heap?



Answer: (B)
A binary tree is max-heap if it is a complete binary tree (A complete binary tree is a binary tree in which every level, except possibly the last, is completely filled, and all nodes are as far left as possible) and it follows the max-heap property (value of each parent is greater than or equal to the values of its children).

A) is not a max-heap because it is not a complete binary tree
B) is a max-heap because it is complete binary tree and follows max-heap property.
C) is not a max-heap because 8 is a chile of 5 in this tree, so violates the max-heap property.
D) is not a max-heap because 8 is a chile of 5 in this tree, so violates the max-heap property. There are many other nodes in this tree which violate max-heap property in this tree.

2) Four matrices M1, M2, M3 and M4 of dimensions pxq, qxr, rxs and sxt respectively can be multiplied is several ways with different number of total scalar multiplications. For example, when multiplied as ((M1 X M2) X (M3 X M4)), the total number of multiplications is pqr + rst + prt. When multiplied as (((M1 X M2) X M3) X M4), the total number of scalar multiplications is pqr + prs + pst.

If p = 10, q = 100, r = 20, s = 5 and t = 80, then the number of scalar multiplications needed is
A) 248000
B) 44000
C) 19000
D) 25000

Answer (C)
We get minimum number of multiplications using ((M1 X (M2 X M3)) X M4).

Total number of multiplications = 100x20x5 (for M2 x M3) + 10x100x5 + 10x5x80 =

19000.

3) Which of the given options provides the increasing order of asymptotic complexity of functions f1, f2, f3 and f4?

f1(n) = 2^n
f2(n) = n^(3/2)
f3(n) = nLogn
f4(n) = n^(Logn)

A) f3, f2, f4, f1
B) f3, f2, f1, f4
C) f2, f3, f1, f4
D) f2, f3, f4, f1

Answer (B)

```
f1(n) = 2^n
f2(n) = n^(3/2)
f3(n) = nLogn
f4(n) = n^(Logn)
```

Except f2, all other are exponential. So f2 is definitely first in output. Among remaining, n^(3/2) is next.

Let us compare f4 and f1. Let us take few values to compare

```
n = 32, f1 = 2^32, f4 = 32^5 = 2^25
n = 64, f1 = 2^64, f4 = 64^6 = 2^36
..............
..............
```

Also see http://www.wolframalpha.com/input/?i=2^n+vs+n^%28log+n%29
Thanks to fella26 for suggesting the above explanation.

4) We are given a set of n distinct elements and an unlabeled binary tree with n nodes. In how many ways can we populate the tree with the given set so that it becomes a binary search tree?
A) 0
B) 1
C) n!
D) (1/(n+1)).2nCn

Answer (B)

See this explanation from PeddaBoku.

5) An algorithm to find the length of the longest monotonically increasing sequence of numbers in an array A[0 :n-1] is given below.
Let Li denote the length of the longest monotonically increasing sequence starting at index i in the array

Initialize $L_{n-1} = 1$
For all i such that $0 \le i \le n-2$

$$L_i = \begin{cases} 1 + L_{i+1} & \text{if } A[i] < A[i+1] \\ 1 & \text{Otherwise} \end{cases}$$

Finally the length of the longest monotonically increasing sequence is $\text{Max}(L_0, L_1, \ldots, L_{n-1})$.

Which of the following statements is TRUE?
(A) The algorithm uses dynamic programming paradigm
(B) The algorithm has a linear complexity and uses branch and bound paradigm
(C) The algorithm has a non-linear polynomial complexity and uses branch and bound paradigm
(D) The algorithm uses divide and conquer paradigm.

Answer: (A)

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

43. C Language | Set 8

Following questions have been asked in GATE CS 2011 exam.

1) What does the following fragment of C-program print?

```
char c[] = "GATE2011";
char *p =c;
printf("%s", p + p[3] - p[1]) ;
```

(A) GATE2011
(B) E2011
(C) 2011
(D) 011

Answer: (C)
See comments for explanation.

```
char c[] = "GATE2011";

 // p now has the base address string "GATE2011"
char *p =c;

// p[3] is 'E' and p[1] is 'A'.
// p[3] - p[1] = ASCII value of 'E' - ASCII value of 'A' = 4
// So the expression  p + p[3] - p[1] becomes p + 4 which is
// base address of string "2011"
printf("%s", p + p[3] - p[1]) ;
```

2) Consider the following recursive C function that takes two arguments

```
unsigned int foo(unsigned int n, unsigned int r) {
  if (n  > 0) return (n%r +  foo (n/r, r ));
  else return 0;
}
```

What is the return value of the function foo when it is called as foo(513, 2)?

(A) 9

(B) 8

(C) 5

(D) 2

Answer: (D)

foo(513, 2) will return 1 + foo(256, 2). All subsequent recursive calls (including foo(256, 2)) will return 0 + foo(n/2, 2) except the last call foo(1, 2) . The last call foo(1, 2) returns 1. So, the value returned by foo(513, 2) is 1 + 0 + 0…. + 0 + 1.

The function foo(n, 2) basically returns sum of bits (or count of set bits) in the number n.

3) What is the return value of the function foo when it is called as foo(345, 10) ?

(A) 345

(B) 12

(C) 5

(D) 3

Answer: (B)

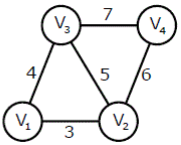
The call foo(345, 10) returns sum of decimal digits (because r is 10) in the number n. Sum of digits for 345 is 3 + 4 + 5 = 12.

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

44. Data Structures and Algorithms | Set 27

Following questions have been asked in GATE CS 2011 exam.

1) An undirected graph G(V, E) contains n (n > 2) nodes named v1 , v2 ,….vn. Two nodes vi , vj are connected if and only if 0 < |i – j| <= 2. Each edge (vi, vj) is assigned a weight i + j. A sample graph with n = 4 is shown below.



What will be the cost of the minimum spanning tree (MST) of such a graph with n nodes?
(A) 1/12(11n^2 – 5n)
(B) n^2 – n + 1
(C) 6n – 11
(D) 2n + 1

Answer: (B)
Minimum spanning tree for 2 nodes would be

```
(v1) _ (v2)
```

Total weight 3

Minimum spanning tree for 3 nodes would be

```
(v1) _ (v2)
   |
(v3)
```

Total weight= 3 + 4 = 7

Minimum spanning tree for 4 nodes would be

```
(v1) _ (v2) _ (v4)
   |
(v3)
```

Total weight= 3 + 4 + 6 = 13

Minimum spanning tree for 5 nodes would be

```
(v1) _ (v2) _ (v4)
   |
(v3)
   |
```

```
(v5)
```

Total weight= 3 + 4 + 6 + 8 = 21

Minimum spanning tree for 6 nodes would be

```
(v1) _ (v2) _ (v4) _ (v6)
   |
(v3)
   |
(v5)
```

Total weight= 3 + 4 + 6 + 8 + 10 = 31

We can observe from above examples that when we add kth node, the weight of spanning tree increases by 2k-2. Let T(n) be the weight of minimum spanning tree. T(n) can be written as

T(n) = T(n-1) + (2n-2) for n > 2
T(1) = 0, T(2) = 0 and T(2) = 3

The recurrence can be written as sum of series (2n – 2) + (2n-4) + (2n-6) + (2n-8) + …. 3 and solution of this recurrence is n^2 – n + 1.

2) The length of the path from v5 to v6 in the MST of previous question with n = 10 is
(A) 11
(B) 25
(C) 31
(D) 41

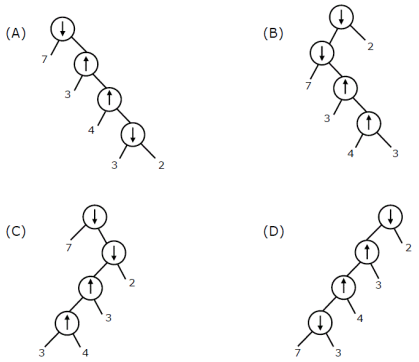
Answer: (C)
Any MST which has more than 5 nodes will have the same distance between v5 and v6 as the basic structure of all MSTs (with more than 5 nodes) would be following.

```
(v1) _ (v2) _ (v4) _  (v6) _ .  . (more even numbered nodes)
   |
(v3)
   |
(v5)
   |
   .
   .
(more odd numbered nodes)
```

Distance between v5 and v6 = 3 + 4 + 6 + 8 + 10 = 31

3) Consider two binary operators ‘↑ ‘ and ‘↓‘ with the precedence of operator ↓

being lower than that of the ↑ operator. Operator ↑ is right associative while operator ↓ is left associative. Which one of the following represents the parse tree for expression (7 ↓ 3 ↑ 4 ↑ 3 ↓ 2)?



Answer: (B)
Let us consider the given expression ($7 \downarrow 3 \uparrow 4 \uparrow 3 \downarrow 2$).

Since the precedence of ↑ is higher, the sub-expression ($3 \uparrow 4 \uparrow 3$) will be evaluated first. In this sub-expression, $4 \uparrow 3$ would be evaluated first because ↑ is right to left associative. So the expression is evaluated as $((7 \downarrow (3 \uparrow (4 \uparrow 3))) \downarrow 2)$. Also, note that among the two ↓ operators, first one is evaluated before the second one because the associativity of ↓ is left to right.

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

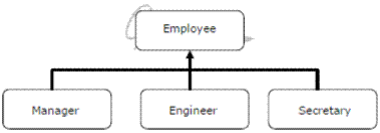
45. Object Oriented Programming | Set 1

Following questions have been asked in GATE CS exam.

1. It is desired to design an object-oriented employee record system for a company. Each employee has a name, unique id and salary. Employees belong to different categories and their salary is determined by their category. The functions to get Name, getId and compute salary are required. Given the class hierarchy below, possible locations for these functions are: (GATE CS 2004)

i. getId is implemented in the superclass
ii. getId is implemented in the subclass
iii. getName is an abstract function in the superclass
iv. getName is implemented in the superclass

v. getName is implemented in the subclass
vi. getSalary is an abstract function in the superclass
vii. getSalary is implemented in the superclass
viii. getSalary is implemented in the subclass



Choose the best design

(a) (i), (iv), (vi), (viii)
(b) (i), (iv), (vii)
(c) (i), (iii), (v), (vi), (viii)
(d) (ii), (v), (viii)

Answer (a)
Getid() and GetName() can be there in the base class as these functions have the same implementation for all subclasses. As the question says that every employee must have salary and salary is determined by their category, getSalary() must be there as an abstract function in base class. And all subclasses should implement salary according to their category.

2. Which one of the following are essential features of an object-oriented programming language? (GATE CS 2005)
(i) Abstraction and encapsulation
(ii) Strictly-typedness
(iii) Type-safe property coupled with sub-type rule
(iv) Polymorphism in the presence of inheritance

(a) (i) and (ii) only
(b) (i) and (iv) only
(c) (i), (ii) and (iv) only
(d) (i), (iii) and (iv) only

Answer (b)
Abstraction, Encapsulation, Polymorphism and Inheritance are the essential features of a OOP Language (See the Wiki page for OOP).

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

46. Data Structures and Algorithms | Set 28

Following questions have been asked in GATE 2012 exam.

1) Let w(n) and A(n) denote respectively, the worst case and average case running time of an algorithm executed on an input of size n. which of the following is ALWAYS TRUE?

(A) $A(n) = \Omega(W(n))$
(B) $A(n) = \Theta(W(n))$
(C) $A(n) = O(W(n))$
(D) $A(n) = o(W(n))$

Answer (C)
The worst case time complexity is always greater than or same as the average case time complexity.

2) The worst case running time to search for an element in a balanced in a binary search tree with n2^n elements is

(A) $\Theta(nlogn)$
(B) $\Theta(n2^n)$
(C) $\Theta(n)$
(D) $\Theta(logn)$

Answer (C)
Time taken to search an element is $\Theta(h)$ where h is the height of Binary Search Tree (BST). The growth of height of a balanced BST is logerthimic in terms of number of nodes. So the worst case time to search an element would be $\Theta(Log(n * 2^n))$ which is $\Theta(Log(n) + Log(2^n))$ Which is $\Theta(Log(n) + n)$ which can be written as $\Theta(n)$.

3) Assuming P != NP, which of the following is true ?

(A) NP-complete = NP
(B) NP-complete $\cap$ P = Φ
(C) NP-hard = NP
(D) P = NP-complete

Answer (B)
The answer is B (no NP-Complete problem can be solved in polynomial time). Because, if one NP-Complete problem can be solved in polynomial time, then all NP problems can solved in polynomial time. If that is the case, then NP and P set become same which contradicts the given condition.

4) The height of a tree is defined as the number of edges on the longest path in the tree. The function shown in the pseudocode below is invoked as height (root) to

compute the height of a binary tree rooted at the tree pointer root.

```
int height (treeptr n)
{ if (n== NULL) return -1;
  if (n → left == NULL)
        if (n → right ==NULL) return 0;
        else return B1;                                  // Box 1
else {h1 = height (n → left);
        if (n → right == NULL) return (1 + h1);
        else {h2 = height (n → right);
                return B2 ;                              // Box 2
                }
        }
}
```

The appropriate expression for the two boxes B1 and B2 are

(A) B1 : (1 + height(n->right)), B2 : (1 + max(h1,h2))

(B) B1 : (height(n->right)), B2 : (1 + max(h1,h2))

(C) B1 : height(n->right), B2 : max(h1,h2)

(D) B1 : (1 + height(n->right)), B2 : max(h1,h2)

Answer (A)

The box B1 gets exected when left subtree of n is NULL and right sbtree is not NULL. In this case, height of n will be height of right subtree plus one.

The box B2 gets executed when both left and right sbtrees of n are not NULL. In this case, height of n will be max of heights of left and right sbtrees of n plus 1.

5) A list of n string, each of length n, is sorted into lexicographic order using the merge-sort algorithm. The worst case running time of this computation is

(A) $O(nlogn$

(B) $O(n^2logn)$

(C) $O(n^2 + logn)$

(D) $O(n^2)$

Answer (B)

The recurrence tree for merge sort will have height Logn. And O(n^2) work will be done at each level of the recurrence tree (Each level involves n comparisons and a comparison takes O(n) time in worst case). So time complexity of this Merge Sort will be $O(n^2logn)$.

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

47. C Language | Set 9

Following questions have been asked in GATE 2012 exam.

1. What will be the output of the following C program segment?

```
char inchar = 'A';
switch (inchar)
{
case 'A' :
    printf ("choice A \n") ;
case 'B' :
    printf ("choice B ") ;
case 'C' :
case 'D' :
case 'E' :
default:
    printf ("No Choice") ;
}
```

(A) No choice
(B) Choice A
(C) Choice A
Choice B No choice
(D) Program gives no output as it is erroneous

Answer (C)
There is no break statement in case 'A'. If a case is executed and it doesn't contain break, then all the subsequent cases are executed until a break statement is found. That is why everything inside the switch is printed.
Try following program as an exercise.

```
int main()
{
    char inchar = 'A';
    switch (inchar)
    {
    case 'A' :
        printf ("choice A \n") ;
    case 'B' :
    {
        printf ("choice B") ;
        break;
    }
    case 'C' :
    case 'D' :
    case 'E' :
    default:
        printf ("No Choice") ;
    }
}
```

2. Consider the following C program

```
int a, b, c = 0;
void prtFun (void);
int main ()
{
    static int a = 1; /* line 1 */
    prtFun();
    a += 1;
    prtFun();
    printf ( "\n %d %d " , a, b) ;
}

void prtFun (void)
{
    static int a = 2; /* line 2 */
    int b = 1;
    a += ++b;
    printf (" \n %d %d " , a, b);
}
```

What output will be generated by the given code segment?

(A) 3 1
4 1
4 2
(B) 4 2
6 1
6 1
(C) 4 2
6 2
2 0
(D) 3 1
5 2
5 2

Answer (C)
'a' and 'b' are global variable. prtFun() also has 'a' and 'b' as local variables. The local variables hide the globals (See Scope rules in C). When prtFun() is called first time, the local 'b' becomes 2 and local 'a' becomes 4.
When prtFun() is called second time, same instance of local static 'a' is used and a new instance of 'b' is created because 'a' is static and 'b' is non-static. So 'b' becomes 2 again and 'a' becomes 6.
main() also has its own local static variable named 'a' that hides the global 'a' in main. The printf() statement in main() accesses the local 'a' and prints its value. The same printf() statement accesses the global 'b' as there is no local variable named 'b' in main. Also, the defaut value of static and global int variables is 0. That is why the printf statement in main() prints 0 as value of b.

3. What output will be generated by the given code d\segment if:
Line 1 is replaced by "auto int a = 1;"
Line 2 is replaced by "register int a = 2;"
(A) 3 1

4 1
4 2
(B) 4 2
6 1
6 1
(C) 4 2
6 2
2 0
(D) 4 2
4 2
2 0

Answer (D)
If we replace line 1 by “auto int a = 1;” and line 2 by “register int a = 2;”, then ‘a’ becomes non-static in prtFun(). The output of first prtFun() remains same. But, the output of second prtFun() call is changed as a new instance of ‘a’ is created in second call. So “4 2″ is printed again. Finally, the printf() in main will print “2 0″. Making ‘a’ a register variable won’t change anything in output.

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

48. Data Structures and Algorithms | Set 29

Following questions have been asked in GATE 2012 exam.

1) The recurrence relation capturing the optimal time of the Tower of Hanoi problem with n discs is
(A) T(n) = 2T(n – 2) + 2
(B) T(n) = 2T(n – 1) + n
(C) T(n) = 2T(n/2) + 1
(D) T(n) = 2T(n – 1) + 1

Answer (D)

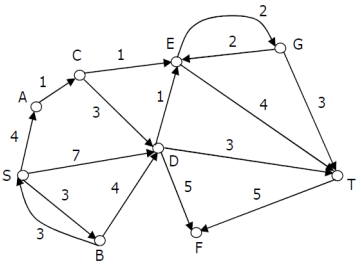
Following are the steps to follow to solve Tower of Hanoi problem recursively.

```
Let the three pegs be A, B and C. The goal is to move n pegs from A to C.
To move n discs from peg A to peg C:
    move n-1 discs from A to B. This leaves disc n alone on peg A
    move disc n from A to C
    move n?1 discs from B to C so they sit on disc n
```

The recurrence function T(n) for time complexity of the above recursive solution can be written as following.

T(n) = 2T(n-1) + 1

2) Consider the directed graph shown in the figure below. There are multiple shortest paths between vertices S and T. Which one will be reported by Dijstra?s shortest path algorithm? Assume that, in any iteration, the shortest path to a vertex v is updated only when a strictly shorter path to v is discovered.



(A) SDT
(B) SBDT
(C) SACDT
(D) SACET

Answer (D)

3) Suppose a circular queue of capacity (n – 1) elements is implemented with an array of n elements. Assume that the insertion and deletion operation are carried out using REAR and FRONT as array index variables, respectively. Initially, REAR = FRONT = 0. The conditions to detect queue full and queue empty are
(A) Full: (REAR+1) mod n == FRONT, empty: REAR == FRONT
(B) Full: (REAR+1) mod n == FRONT, empty: (FRONT+1) mod n == REAR
(C) Full: REAR == FRONT, empty: (REAR+1) mod n == FRONT
(D) Full: (FRONT+1) mod n == REAR, empty: REAR == FRONT

Answer (A)
See this for details.

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

49. Operating Systems | Set 5

Following questions have been asked in GATE 2012 exam.

1. A process executes the code

```
fork ();
fork ();
fork ();
```

The total number of child processes created is
(A) 3
(B) 4
(C) 7
(D) 8

Answer (C)

Let us put some label names for the three lines

```
fork ();    // Line 1
fork ();   // Line 2
fork ();   // Line 3

      L1       // There will be 1 child process created by line 1
   /     \
  L2      L2    // There will be 2 child processes created by line 2
 /  \    /  \
L3  L3  L3  L3  // There will be 4 child processes created by line 3
```

We can also use direct formula to get the number of child processes. With n fork statements, there are always 2^n – 1 child processes. Also see this post for more details.

2. consider the 3 processes, P1, P2 and P3 shown in the table

Process	Arrival time	Time unit required
P1	0	5
P2	1	7
P3	3	4

The completion order of the 3 processes under the policies FCFS and RRS (round robin scheduling with CPU quantum of 2 time units) are
(A) **FCFS**: P1, P2, P3 **RR2**: P1, P2, P3
(B) **FCFS**: P1, P3, P2 **RR2**: P1, P3, P2
(C) **FCFS**: P1, P2, P3 **RR2**: P1, P3, P2
(D) **FCFS**: P1, P3, P2 **RR2**: P1, P2, P3

Answer (C)

3. Consider the virtual page reference string
1, 2, 3, 2, 4, 1, 3, 2, 4, 1
On a demand paged virtual memory system running on a computer system that main memory size of 3 pages frames which are initially empty. Let LRU, FIFO and OPTIMAL denote the number of page faults under the corresponding page replacements policy. Then
(A) OPTIMAL < LRU < FIFO
(B) OPTIMAL < FIFO < LRU
(C) OPTIMAL = LRU
(D) OPTIMAL = FIFO

Answer (B)
The OPTIMAL will be 5, FIFO 6 and LRU 9.

4. A file system with 300 GByte uses a file descriptor with 8 direct block address. 1 indirect block address and 1 doubly indirect block address. The size of each disk block is 128 Bytes and the size of each disk block address is 8 Bytes. The maximum possible file size in this file system is
(A) 3 Kbytes
(B) 35 Kbytes
(C) 280 Bytes
(D) Dependent on the size of the disk

Answer (B)

Total number of possible addresses stored in a disk block = 128/8 = 16

Maximum number of addressable bytes due to direct address block = 8*128
Maximum number of addressable bytes due to 1 single indirect address block = 16*128
Maximum number of addressable bytes due to 1 double indirect address block = 16*16*128

The maximum possible file size = 8*128 + 16*128 + 16*16*128 = 35KB

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

50. Operating Systems | Set 6

Following questions have been asked in GATE 2011 CS exam.

1) A thread is usually defined as a 'light weight process' because an operating system (OS) maintains smaller data structures for a thread than for a process. In relation to this, which of the followings is TRUE?
(A) On per-thread basis, the OS maintains only CPU register state
(B) The OS does not maintain a separate stack for each thread
(C) On per-thread basis, the OS does not maintain virtual memory state
(D) On per thread basis, the OS maintains only scheduling and accounting information.

Answer (A)
Threads are called 'light weight process' because they only need storage for stack and registers. They don't need separate space for other things like code segment, global data, etc

2) Let the page fault service time be 10ms in a computer with average memory access time being 20ns. If one page fault is generated for every 10^6 memory accesses, what is the effective access time for the memory?
(A) 21ns
(B) 30ns
(C) 23ns
(D) 35ns

Answer (B)

```
Let P be the page fault rate
Effective Memory Access Time = p * (page fault service time) +
                               (1 - p) * (Memory access time)
                             = ( 1/(10^6) )* 10 * (10^6) ns +
                               (1 - 1/(10^6)) * 20 ns
                             = 30 ns (approx)
```

3) An application loads 100 libraries at startup. Loading each library requires exactly one disk access. The seek time of the disk to a random location is given as 10ms. Rotational speed of disk is 6000rpm. If all 100 libraries are loaded from random locations on the disk, how long does it take to load all libraries? (The time to transfer data from the disk block once the head has been positioned at the start of the block may be neglected)
(A) 0.50s
(B) 1.50s
(C) 1.25s
(D) 1.00s

Answer (B)
Since transfer time can be neglected, the average access time is sum of average seek time and average rotational latency. Average seek time for a random location time is given as 10 ms. The average rotational latency is half of the time needed for complete

rotation. It is given that 6000 rotations need 1 minute. So one rotation will take 60/6000 seconds which is 10 ms. Therefore average rotational latency is half of 10 ms, which is 5ms.

```
Average disk access time = seek time + rotational latency
                         = 10 ms + 5 ms
                         = 15 ms
For 100 libraries, the average disk access time will be 15*100 ms
```

4. Consider the following table of arrival time and burst time for three processes P0, P1 and P2.

Process	Arrival time	Burst Time
P0	0 ms	9 ms
P1	1 ms	4 ms
P2	2 ms	9 ms

The pre-emptive shortest job first scheduling algorithm is used. Scheduling is carried out only at arrival or completion of processes. What is the average waiting time for the three processes?
(A) 5.0 ms
(B) 4.33 ms
(C) 6.33 ms
(D) 7.33 ms

Answer: – (A)
Process P0 is allocated processor at 0 ms as there is no other process in ready queue. P0 is preempted after 1 ms as P1 arrives at 1 ms and burst time for P1 is less than remaining time of P0. P1 runs for 4ms. P2 arrived at 2 ms but P1 continued as burst time of P2 is longer than P1. After P1 completes, P0 is scheduled again as the remaining time for P0 is less than the burst time of P2.
P0 waits for 4 ms, P1 waits for 0 ms amd P2 waits for 11 ms. So average waiting time is (0+4+11)/3 = 5.

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

51. Operating Systems | Set 7

Following questions have been asked in GATE CS exam.

1) Let the time taken to switch between user and kernel modes of execution be t1 while the time taken to switch between two processes be t2. Which of the following is TRUE? (GATE CS 2011)
(A) t1 > t2
(B) t1 = t2
(C) t1 < t2
(D) Nothing can be said about the relation between t1 and t2

Answer: - (C)
Process switching involves mode switch. Context switching can occur only in kernel mode.

2) A system uses FIFO policy for page replacement. It has 4 page frames with no pages loaded to begin with. The system first accesses 100 distinct pages in some order and then accesses the same 100 pages but now in the reverse order. How many page faults will occur? (GATE CS 2010)

(A) 196
(B) 192
(C) 197
(D) 195

Answer (A)
Access to 100 pages will cause 100 page faults. When these pages are accessed in reverse order, the first four accesses will not cause page fault. All other access to pages will cause page faults. So total number of page faults will be 100 + 96.

3) Which of the following statements are true? (GATE CS 2010)
I. Shortest remaining time first scheduling may cause starvation
II. Preemptive scheduling may cause starvation
III. Round robin is better than FCFS in terms of response time
(A) I only
(B) I and III only
(C) II and III only
(D) I, II and III

Answer (D)

I) Shortest remaining time first scheduling is a preemptive version of shortest job scheduling. It may cause starvation as shorter processes may keep coming and a long CPU burst process never gets CPU.
II) Preemption may cause starvation. If priority based scheduling with preemption is used, then a low priority process may never get CPU.
III) Round Robin Scheduling improves response time as all processes get CPU after a specified time.

4) Consider the methods used by processes P1 and P2 for accessing their critical sections whenever needed, as given below. The initial values of shared boolean variables S1 and S2 are randomly assigned.

```
Method Used by P1
while (S1 == S2) ;
Critical Section
S1 = S2;

Method Used by P2
while (S1 != S2) ;
Critical Section
S2 = not (S1);
```

Which one of the following statements describes the properties achieved? (GATE CS 2010)
(A) Mutual exclusion but not progress
(B) Progress but not mutual exclusion
(C) Neither mutual exclusion nor progress
(D) Both mutual exclusion and progress

Answer (A)
It can be easily observed that the Mutual Exclusion requirement is satisfied by the above solution, P1 can enter critical section only if S1 is not equal to S2, and P2 can enter critical section only if S1 is equal to S2.
Progress Requirement is not satisfied. Let us first see definition of Progress Requirement.
Progress Requirement: If no process is executing in its critical section and there exist some processes that wishes to enter their critical section, then the selection of the processes that will enter the critical section next cannot be postponed indefinitely.
If P1 or P2 want to re-enter the critical section, then they cannot even if there is other process running in critical section.

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

52. Operating Systems | Set 8

Following questions have been asked in GATE 2009 CS exam.

1) In which one of the following page replacement policies, Belady's anomaly may

occur?
(A) FIFO
(B) Optimal
(C) LRU
(D) MRU

Answer (A)
Belady's anomaly proves that it is possible to have more page faults when increasing the number of page frames while using the First in First Out (FIFO) page replacement algorithm.
See the wiki page for an example of increasing page faults with number of page frames.

2) The essential content(s) in each entry of a page table is / are
(A) Virtual page number
(B) Page frame number
(C) Both virtual page number and page frame number
(D) Access right information

Answer (B)
A page table entry must contain Page frame number. Virtual page number is typically used as index in page table to get the corresponding page frame number. See this for details.

3) Consider a system with 4 types of resources R1 (3 units), R2 (2 units), R3 (3 units), R4 (2 units). A non-preemptive resource allocation policy is used. At any given instance, a request is not entertained if it cannot be completely satisfied. Three processes P1, P2, P3 request the sources as follows if executed independently.

```
Process P1:
t=0: requests 2 units of R2
t=1: requests 1 unit of R3
t=3: requests 2 units of R1
t=5: releases 1 unit of R2
        and 1 unit of R1.
t=7: releases 1 unit of R3
t=8: requests 2 units of R4
t=10: Finishes

Process P2:
t=0: requests 2 units of R3
t=2: requests 1 unit of R4
t=4: requests 1 unit of R1
t=6: releases 1 unit of R3
```

```
t=8: Finishes

Process P3:
t=0: requests 1 unit of R4
t=2: requests 2 units of R1
t=5: releases 2 units of R1
t=7: requests 1 unit of R2
t=8: requests 1 unit of R3
t=9: Finishes
```

Which one of the following statements is TRUE if all three processes run concurrently starting at time t=0?
(A) All processes will finish without any deadlock
(B) Only P1 and P2 will be in deadlock.
(C) Only P1 and P3 will be in a deadlock.
(D) All three processes will be in deadlock

Answer (A)
We can apply the following Deadlock Detection algorithm and see that there is no process waiting indefinitely for a resource. See this for deadlock detection algorithm.

4) Consider a disk system with 100 cylinders. The requests to access the cylinders occur in following sequence:
4, 34, 10, 7, 19, 73, 2, 15, 6, 20
Assuming that the head is currently at cylinder 50, what is the time taken to satisfy all requests if it takes 1ms to move from one cylinder to adjacent one and shortest seek time first policy is used?
(A) 95ms
(B) 119ms
(C) 233ms
(D) 276ms

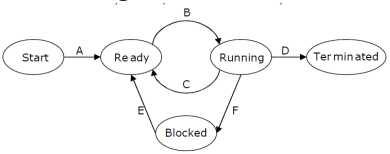
Answer (B)
4, 34, 10, 7, 19, 73, 2, 15, 6, 20
Since shortest seek time first policy is used, head will first move to 34. This move will cause 16*1 ms. After 34, head will move to 20 which will cause 14*1 ms. And so on. So cylinders are accessed in following order 34, 20, 19, 15, 10, 7, 6, 4, 2, 73 and total time will be (16 + 14 + 1 + 4 + 5 + 3 + 1 + 2 + 2 + 71)*1 = 119 ms.

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

53. Operating Systems | Set 9

Following questions have been asked in GATE 2009 CS exam.

1) In the following process state transition diagram for a uniprocessor system, assume that there are always some processes in the ready state: Now consider the following statements:



I. If a process makes a transition D, it would result in another process making transition A immediately.
II. A process P2 in blocked state can make transition E while another process P1 is in running state.
III. The OS uses preemptive scheduling.
IV. The OS uses non-preemptive scheduling.
Which of the above statements are TRUE?
(A) I and II
(B) I and III
(C) II and III
(D) II and IV

Answer (C)
I is false. If a process makes a transition D, it would result in another process making transition B, not A.
II is true. A process can move to ready state when I/O completes irrespective of other process being in running state or not.
III is true because there is a transition from running to ready state.
IV is false as the OS uses preemptive scheduling.

2) The enter_CS() and leave_CS() functions to implement critical section of a process are realized using test-and-set instruction as follows:

```
void enter_CS(X)
{
    while test-and-set(X) ;
}
void leave_CS(X)
{
   X = 0;
}
```

In the above solution, X is a memory location associated with the CS and is

initialized to 0. Now consider the following statements:
I. The above solution to CS problem is deadlock-free
II. The solution is starvation free.
III. The processes enter CS in FIFO order.
IV More than one process can enter CS at the same time.
Which of the above statements is TRUE?
(A) I only
(B) I and II
(C) II and III
(D) IV only

Answer (A)
The above solution is a simple test-and-set solution that makes sure that deadlock doesn't occur, but it doesn't use any queue to avoid starvation or to have FIFO order.

3) A multilevel page table is preferred in comparison to a single level page table for translating virtual address to physical address because
(A) It reduces the memory access time to read or write a memory location.
(B) It helps to reduce the size of page table needed to implement the virtual address space of a process.
(C) It is required by the translation lookaside buffer.
(D) It helps to reduce the number of page faults in page replacement algorithms.

Answer (B)
The size of page table may become too big (See this) to fit in contiguous space. That is why page tables are typically divided in levels.

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

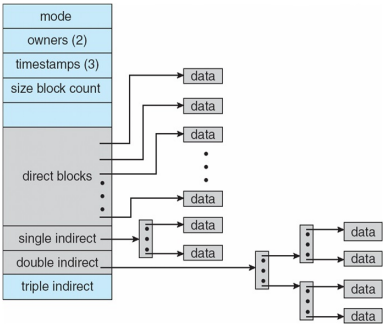
54. Operating Systems | Set 10

Following questions have been asked in GATE 2008 CS exam.

1) The data blocks of a very large file in the Unix file system are allocated using
(A) contiguous allocation
(B) linked allocation
(C) indexed allocation
(D) an extension of indexed allocation

Answer (D)
The Unix file system uses an extension of indexed allocation. It uses direct blocks, single

indirect blocks, double indirect blocks and triple indirect blocks. Following diagram shows implementation of Unix file system. The diagram is taken from Operating System Concept book.



2) The P and V operations on counting semaphores, where s is a counting semaphore, are defined as follows:

```
P(s) : s =  s - 1;
  if (s  < 0) then wait;
V(s) : s = s + 1;
  if (s <= 0) then wakeup a process waiting on s;
```

Assume that Pb and Vb the wait and signal operations on binary semaphores are provided. Two binary semaphores Xb and Yb are used to implement the semaphore operations P(s) and V(s) as follows:

```
P(s) : Pb(Xb);
  s = s - 1;
  if (s < 0) {
   Vb(Xb) ;
   Pb(Yb) ;
  }
  else Vb(Xb);


V(s) : Pb(Xb) ;
  s = s + 1;
  if (s <= 0) Vb(Yb) ;
  Vb(Xb) ;
```

The initial values of Xb and Yb are respectively
(A) 0 and 0
(B) 0 and 1
(C) 1 and 0

(D) 1 and 1

Answer (C)
Both P(s) and V(s) operations are perform Pb(xb) as first step. If Xb is 0, then all processes executing these operations will be blocked. Therefore, Xb must be 1.
If Yb is 1, it may become possible that two processes can execute P(s) one after other (implying 2 processes in critical section). Consider the case when s = 1, y = 1. So Yb must be 0.

3) Which of the following statements about synchronous and asynchronous I/O is NOT true?
(A) An ISR is invoked on completion of I/O in synchronous I/O but not in asynchronous I/O
(B) In both synchronous and asynchronous I/O, an ISR (Interrupt Service Routine) is invoked after completion of the I/O
(C) A process making a synchronous I/O call waits until I/O is complete, but a process making an asynchronous I/O call does not wait for completion of the I/O
(D) In the case of synchronous I/O, the process waiting for the completion of I/O is woken up by the ISR that is invoked after the completion of I/O

Answer (A)
In both Synchronous and Asynchronous, an interrupt is generated on completion of I/O. In Synchronous, interrupt is generated to wake up the process waiting for I/O. In Asynchronous, interrupt is generated to inform the process that the I/O is complete and it can process the data from the I/O operation. See this for more details.

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

55. Operating Systems | Set 11

Following questions have been asked in GATE 2008 CS exam.

1) A process executes the following code

```
for (i = 0; i < n; i++) fork();
```

The total number of child processes created is
(A) n
(B) 2^n - 1
(C) 2^n

(D) 2^(n+1) - 1;

Answer (B)

```
         F0       // There will be 1 child process created by first fork
      /     \
    F1      F1    // There will be 2 child processes created by second fork
   /  \    /  \
 F2   F2  F2   F2  // There will be 4 child processes created by third fork
/ \   / \ / \  / \
 ..............   // and so on
```

If we sum all levels of above tree for i = 0 to n-1, we get 2^n - 1. So there will be 2^n – 1 child processes. Also see this post for more details.

2) Which of the following is NOT true of deadlock prevention and deadlock avoidance schemes?
(A) In deadlock prevention, the request for resources is always granted if the resulting state is safe
(B) In deadlock avoidance, the request for resources is always granted if the result state is safe
(C) Deadlock avoidance is less restrictive than deadlock prevention
(D) Deadlock avoidance requires knowledge of resource requirements a priori

Answer (A)
Deadlock prevention scheme handles deadlock by making sure that one of the four necessary conditions don't occur. In deadlock prevention, the request for a resource may not be granted even if the resulting state is safe. (See the Galvin book slides for more details)

3) A processor uses 36 bit physical addresses and 32 bit virtual addresses, with a page frame size of 4 Kbytes. Each page table entry is of size 4 bytes. A three level page table is used for virtual to physical address translation, where the virtual address is used as follows
• Bits 30-31 are used to index into the first level page table
• Bits 21-29 are used to index into the second level page table
• Bits 12-20 are used to index into the third level page table, and
• Bits 0-11 are used as offset within the page
The number of bits required for addressing the next level page table (or page frame) in the page table entry of the first, second and third level page tables are respectively
(A) 20, 20 and 20
(B) 24, 24 and 24
(C) 24, 24 and 20
(D) 25, 25 and 24

Answer (D)
Virtual address size = 32 bits
Physical address size = 36 bits
Physical memory size = 2^36 bytes
Page frame size = 4K bytes = 2^12 bytes
No. of bits required to access physical memory frame = 36 - 12 = 24
So in third level of page table, 24 bits are required to access an entry.

9 bits of virtual address are used to access second level page table entry and size of pages in second level is 4 bytes. So size of second level page table is (2^9)*4 = 2^11 bytes. It means there are (2^36)/(2^11) possible locations to store this page table. Therefore the second page table requires 25 bits to address it. Similarly, the third page table needs 25 bits to address it.

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

56. Operating Systems | Set 12

Following questions have been asked in GATE CS 2007 exam.

1) Consider a disk pack with 16 surfaces, 128 tracks per surface and 256 sectors per track. 512 bytes of data are stored in a bit serial manner in a sector. The capacity of the disk pack and the number of bits required to specify a particular sector in the disk are respectively:
(A) 256 Mbyte, 19 bits
(B) 256 Mbyte, 28 bits
(C) 512 Mbyte, 20 bits
(D) 64 Gbyte, 28 bits

Answer (A)
Capacity of the disk = 16 surfaces X 128 tracks X 256 sectors X 512 bytes = 256 Mbytes.
To calculate number of bits required to access a sector, we need to know total number of sectors. Total number of sectors = 16 surfaces X 128 tracks X 256 sectors = 2^19
So the number of bits required to access a sector is 19.

2) Group 1 contains some CPU scheduling algorithms and Group 2 contains some applications. Match entries in Group 1 to entries in Group 2.

```
    Group I                          Group II
```

```
(P) Gang Scheduling              (1) Guaranteed Scheduling
(Q) Rate Monotonic Scheduling    (2) Real-time Scheduling
(R) Fair Share Scheduling        (3) Thread Scheduling
```

(A) P – 3 Q – 2 R – 1
(B) P – 1 Q – 2 R – 3
(C) P – 2 Q – 3 R – 1
(D) P – 1 Q – 3 R – 2

Answer (A)
Gang scheduling for parallel systems that schedules related threads or processes to run simultaneously on different processors.
Rate monotonic scheduling is used in real-time operating systems with a static-priority scheduling class. The static priorities are assigned on the basis of the cycle duration of the job: the shorter the cycle duration is, the higher is the job's priority.
Fair Share Scheduling is a scheduling strategy in which the CPU usage is equally distributed among system users or groups, as opposed to equal distribution among processes. It is also known as Guaranteed scheduling.

3) An operating system uses Shortest Remaining Time first (SRT) process scheduling algorithm. Consider the arrival times and execution times for the following processes:

```
Process  Execution time  Arrival time
P1             20            0
P2             25            15
P3             10            30
P4             15            45
```

What is the total waiting time for process P2?
(A) 5
(B) 15
(C) 40
(D) 55

Answer (B)
At time 0, P1 is the only process, P1 runs for 15 time units.
At time 15, P2 arrives, but P1 has the shortest remaining time. So P1 continues for 5 more time units.
At time 20, P2 is the only process. So it runs for 10 time units
At time 30, P3 is the shortest remaining time process. So it runs for 10 time units
At time 40, P2 runs as it is the only process. P2 runs for 5 time units.
At time 45, P3 arrives, but P2 has the shortest remaining time. So P2 continues for 10 more time units.
P2 completes its ececution at time 55

```
Total waiting time for P2 = Complition time - (Arrival time + Execution time)
                          = 55 - (15 + 25)
                          = 15
```

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above

57. Operating Systems | Set 13

Following questions have been asked in GATE CS 2007 exam.

1) A virtual memory system uses First In First Out (FIFO) page replacement policy and allocates a fixed number of frames to a process. Consider the following statements:
P: Increasing the number of page frames allocated to a process sometimes increases the page fault rate.
Q: Some programs do not exhibit locality of reference. Which one of the following is TRUE?
(A) Both P and Q are true, and Q is the reason for P
(B) Both P and Q are true, but Q is not the reason for P.
(C) P is false, but Q is true
(D) Both P and Q are false.

Answer (B)
P is true. Increasing the number of page frames allocated to process may increases the no. of page faults (See Belady's Anomaly).
Q is also true, but Q is not the reason for-P as Belady's Anomaly occurs for some specific patterns of page references.

2) A single processor system has three resource types X, Y and Z, which are shared by three processes. There are 5 units of each resource type. Consider the following scenario, where the column alloc denotes the number of units of each resource type allocated to each process, and the column request denotes the number of units of each resource type requested by a process in order to complete execution. Which of these processes will finish LAST?

```
    alloc          request
    X Y Z           X Y Z
P0  1 2 1           1 0 3
```

```
P1   2 0 1            0 1 2
P2   2 2 1            1 2 0
```

(A) P0
(B) P1
(C) P2
(D) None of the above, since the system is in a deadlock

Answer (C)
Once all resources (5, 4 and 3 instances of X, Y and Z respectively) are allocated, 0, 1 and 2 instances of X, Y and Z are left. Only needs of P1 can be satisfied. So P1 can finish its execution first. Once P1 is done, it releases 2, 1 and 3 units of X, Y and Z respectively. Among P0 and P2, needs of P0 can only be satisfied. So P0 finishes its execution. Finally, P2 finishes its execution.

3) Two processes, P1 and P2, need to access a critical section of code. Consider the following synchronization construct used by the processes:Here, wants1 and wants2 are shared variables, which are initialized to false. Which one of the following statements is TRUE about the above construct?

```
  /* P1 */
while (true) {
  wants1 = true;
  while (wants2 == true);
  /* Critical
    Section */
  wants1=false;
}
/* Remainder section */


/* P2 */
while (true) {
  wants2 = true;
  while (wants1==true);
  /* Critical
    Section */
  wants2 = false;
}
/* Remainder section */
```

(A) It does not ensure mutual exclusion.
(B) It does not ensure bounded waiting.
(C) It requires that processes enter the critical section in strict alternation.
(D) It does not prevent deadlocks, but ensures mutual exclusion.

Answer (D)

The above synchronization constructs don't prevent deadlock. When both wants1 and wants2 become true, both P1 and P2 stuck forever in their while loops waiting for each other to finish.

4) Consider the following statements about user level threads and kernel level threads. Which one of the following statement is FALSE?
(A) Context switch time is longer for kernel level threads than for user level threads.
(B) User level threads do not need any hardware support.
(C) Related kernel level threads can be scheduled on different processors in a multi-processor system.
(D) Blocking one kernel level thread blocks all related threads.

Answer (D)
Since kernel level threads are managed by kernel, blocking one thread doesn't cause all related threads to block. It's a problem with user level threads. See this for more details.

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above

58. Operating Systems | Set 14

Following questions have been asked in GATE CS 2006 exam.

1) Consider three CPU-intensive processes, which require 10, 20 and 30 time units and arrive at times 0, 2 and 6, respectively. How many context switches are needed if the operating system implements a shortest remaining time first scheduling algorithm? Do not count the context switches at time zero and at the end.
(A) 1
(B) 2
(C) 3
(D) 4

Answer (B)
Let three process be P0, P1 and P2 with arrival times 0, 2 and 6 respectively and CPU burst times 10, 20 and 30 respectively. At time 0, P0 is the only available process so it runs. At time 2, P1 arrives, but P0 has the shortest remaining time, so it continues. At time 6, P2 arrives, but P0 has the shortest remaining time, so it continues. At time 10, P1 is scheduled as it is the shortest remaining time process. At time 30, P2 is scheduled. Only two context switches are needed. P0 to P1 and P1 to P2.

2) A computer system supports 32-bit virtual addresses as well as 32-bit physical addresses. Since the virtual address space is of the same size as the physical address space, the operating system designers decide to get rid of the virtual memory entirely. Which one of the following is true?
(A) Efficient implementation of multi-user support is no longer possible
(B) The processor cache organization can be made more efficient now
(C) Hardware support for memory management is no longer needed
(D) CPU scheduling can be made more efficient now

Answer (C)
For supporting virtual memory, special hardware support is needed from Memory Management Unit. Since operating system designers decide to get rid of the virtual memory entirely, hardware support for memory management is no longer needed

3) A CPU generates 32-bit virtual addresses. The page size is 4 KB. The processor has a translation look-aside buffer (TLB) which can hold a total of 128 page table entries and is 4-way set associative. The minimum size of the TLB tag is:
(A) 11 bits
(B) 13 bits
(C) 15 bits
(D) 20 bits

Answer (C)
Size of a page = 4KB = 2^12
Total number of bits needed to address a page frame = 32 – 12 = 20
If there are 'n' cache lines in a set, the cache placement is called n-way set associative. Since TLB is 4 way set associative and can hold total 128 (2^7) page table entries, number of sets in cache = 2^7/4 = 2^5. So 5 bits are needed to address a set, and 15 (20 – 5) bits are needed for tag.

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above

59. Operating Systems | Set 15

Following questions have been asked in GATE CS 2006 exam.

1) Consider three processes (process id 0, 1, 2 respectively) with compute time bursts 2, 4 and 8 time units. All processes arrive at time zero. Consider the longest

remaining time first (LRTF) scheduling algorithm. In LRTF ties are broken by giving priority to the process with the lowest process id. The average turn around time is:
(A) 13 units
(B) 14 units
(C) 15 units
(D) 16 units

Answer (A)
Let the processes be p0, p1 and p2. These processes will be executed in following order.

```
  p2  p1  p2  p1  p2  p0  p1   p2   p0   p1   p2
0   4   5   6   7   8   9   10    11   12   13   14
```

Turn around time of a process is total time between submission of the process and its completion.
Turn around time of p0 = 12 (12-0)
Turn around time of p1 = 13 (13-0)
Turn around time of p2 = 14 (14-0)

Average turn around time is (12+13+14)/3 = 13.

2) Consider three processes, all arriving at time zero, with total execution time of 10, 20 and 30 units, respectively. Each process spends the first 20% of execution time doing I/O, the next 70% of time doing computation, and the last 10% of time doing I/O again. The operating system uses a shortest remaining compute time first scheduling algorithm and schedules a new process either when the running process gets blocked on I/O or when the running process finishes its compute burst. Assume that all I/O operations can be overlapped as much as possible. For what percentage of time does the CPU remain idle?(A) 0%
(B) 10.6%
(C) 30.0%
(D) 89.4%

Answer (B)
Let three processes be p0, p1 and p2. Their execution time is 10, 20 and 30 respectively. p0 spends first 2 time units in I/O, 7 units of CPU time and finally 1 unit in I/O. p1 spends first 4 units in I/O, 14 units of CPU time and finally 2 units in I/O. p2 spends first 6 units in I/O, 21 units of CPU time and finally 3 units in I/O.

```
 idle   p0    p1     p2    idle
0    2     9     23     44     47

Total time spent = 47
Idle time = 2 + 3 = 5
Percentage of idle time = (5/47)*100 = 10.6 %
```

3) The atomic *fetch-and-set x, y* instruction unconditionally sets the memory location x to 1 and fetches the old value of x in y without allowing any intervening access to the memory location x. consider the following implementation of P and V functions on a binary semaphore .

```
void P (binary_semaphore *s) {
  unsigned y;
  unsigned *x = &(s->value);
  do {
     fetch-and-set x, y;
  } while (y);
}

void V (binary_semaphore *s) {
  S->value = 0;
}
```

Which one of the following is true?
(A) The implementation may not work if context switching is disabled in P.
(B) Instead of using fetch-and-set, a pair of normal load/store can be used
(C) The implementation of V is wrong
(D) The code does not implement a binary semaphore

Answer (A)
Let us talk about the operation P(). It stores the value of s in x, then it fetches the old value of x, stores it in y and sets x as 1. The while loop of a process will continue forever if some other process doesn't execute V() and sets the value of s as 0. If context switching is disabled in P, the while loop will run forever as no other process will be able to execute V().

4) Consider the following snapshot of a system running n processes. Process i is holding Xi instances of a resource R, 1 <= i <= n. currently, all instances of R are occupied. Further, for all i, process i has placed a request for an additional Yi instances while holding the Xi instances it already has. There are exactly two processes p and q such that Yp = Yq = 0. Which one of the following can serve as a necessary condition to guarantee that the system is not approaching a deadlock?
(A) min (Xp, Xq) < max (Yk) where k != p and k != q
(B) Xp + Xq >= min (Yk) where k != p and k != q
(C) max (Xp, Xq) > 1
(D) min (Xp, Xq) > 1

Answer (B)
Since both p and q don't need additional resources, they both can finish and release Xp + Xq resources without asking for any additional resource. If the resources released by

p and q are sufficient for another process waiting for Yk resources, then system is not approaching deadlock.

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above

60. Database Management Systems | Set 2

Following Questions have been asked in GATE 2012 exam.

1) Which of the following statements are TRUE about an SQL query?
P: An SQL query can contain a HAVING clause even if it does not a GROUP BY clause
Q: An SQL query can contain a HAVING clause only if it has a GROUP BY clause
R: All attributes used in the GROUP BY clause must appear in the SELECT clause
S: Not all attributes used in the GROUP BY clause need to apper in the SELECT clause
(A) P and R
(B) P and S
(C) Q and R
(D) Q and S

Answer (B)
P is correct. HAVING clause can also be used with aggregate function. If we use a HAVING clause without a GROUP BY clause, the HAVING condition applies to all rows that satisfy the search condition. In other words, all rows that satisfy the search condition make up a single group. See this for more details.

S is correct. To verify S, try following queries in SQL.

```
CREATE TABLE temp
  (
     id   INT,
     name VARCHAR(100)
  );

INSERT INTO temp VALUES (1, "abc");
INSERT INTO temp VALUES (2, "abc");
INSERT INTO temp VALUES (3, "bcd");
INSERT INTO temp VALUES (4, "cde");

SELECT Count(*)
FROM   temp
GROUP  BY name;
```

Output:

```
count(*)
--------
2
1
1
```

2) Given the basic ER and relational models, which of the following is INCORRECT?
(A) An attributes of an entity can have more that one value
(B) An attribute of an entity can be composite
(C) In a row of a relational table, an attribute can have more than one value
(D) In a row of a relational table, an attribute can have exactly one value or a NULL value

Answer (C)
The term 'entity' belongs to ER model and the term 'relational table' belongs to relational model.
A and B both are true. ER model supports both multivalued and composite attributes
See this for more details.
(C) is false and (D) is true. In Relation model, an entry in relational table can can have exactly one value or a NULL.

3) Suppose (A, B) and (C,D) are two relation schemas. Let r1 and r2 be the corresponding relation instances. B is a foreign key that refers to C in r2. If data in r1 and r2 satisfy referential integrity constraints, which of the following is ALWAYS TRUE?
(A) $\Pi_B(r_1) - \Pi_C(r_2) = \emptyset$
(B) $\Pi_C(r_2) - \Pi_B(r_1) = \emptyset$
(C) $\Pi_B(r_1) = \Pi_C(r_2)$
(D) $\Pi_B(r_1) - \Pi_C(r_2) \neq \emptyset$

Answer (A)
B is a foreign key in r1 that refers to C in r2. r1 and r2 satisfy referential integrity constraints. So every value that exists in column B of r1 must also exist in column C of r2.

4) Which of the following is TRUE?
(A) Every relation in 2NF is also in BCNF
(B) A relation R is in 3NF if every non-prime attribute of R is fully functionally dependent on every key of R
(C) Every relation in BCNF is also in 3NF
(D) No relation can be in both BCNF and 3NF

Answer (C)

BCNF is a stronger version 3NF. So every relation in BCNF will also be in 3NF.

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above

61. Database Management Systems | Set 3

Following Questions have been asked in GATE 2012 exam.

1) Consider the following transactions with data items P and Q initialized to zero:

```
T1: read (P) ;
    read (Q) ;
    if P = 0 then Q : = Q + 1 ;
    write (Q) ;
T2: read (Q) ;
    read (P) ;
    if Q = 0 then P : = P + 1 ;
    write (P) ;
```

Any non-serial interleaving of T1 and T2 for concurrent execution leads to
(A) A serializable schedule
(B) A schedule that is not conflict serializable
(C) A conflict serializable schedule
(D) A schedule for which a precedence graph cannot be drawn

Answer (B)
Two or more actions are said to be in conflict if:
1) The actions belong to different transactions.
2) At least one of the actions is a write operation.
3) The actions access the same object (read or write).

The schedules S1 and S2 are said to be conflict-equivalent if the following conditions are satisfied:
1) Both schedules S1 and S2 involve the same set of transactions (including ordering of actions within each transaction).
2) The order of each pair of conflicting actions in S1 and S2 are the same.

A schedule is said to be conflict-serializable when the schedule is conflict-equivalent to one or more serial schedules.

Source: Wiki Page for Schedule

In the given scenario, there are two possible serial schedules:

1) T1 followed by T2
2) T2 followed by T1.
In both of the serial schedules, one of the transactions reads the value written by other transaction as a first step. Therefore, any non-serial interleaving of T1 and T2 will not be conflict serializable.

2) Consider the following relations A, B, C. How many tuples does the result of the following relational algebra expression contain? Assume that the schema of A U B is the same as that of A.

$(A \cup B) \rhd\lhd_{A.Id>40 \ \vee \ C.Id<15} C$

Table A

Id	Name	Age
12	Arun	60
15	Shreya	24
99	Rohit	11

Table B

Id	Name	Age
15	Shreya	24
25	Hari	40
98	Rohit	20
99	Rohit	11

Table C

Id	Phone	Area
10	2200	02
99	2100	01

(A) 7
(B) 4
(C) 5
(D) 9

Answer (A)

Result of AUB will be following table

Id	Name	Age

```
12   Arun    60
15   Shreya  24
99   Rohit   11
25   Hari    40
98   Rohit   20

The result of given relational algebra expression will be

Id   Name    Age  Id   Phone Area
---------------------------------
12   Arun    60   10   2200   02
15   Shreya  24   10   2200   02
99   Rohit   11   10   2200   02
25   Hari    40   10   2200   02
98   Rohit   20   10   2200   02
99   Rohit   11   99   2100   01
98   Rohit   20   99   2100   01
```

3) Consider the above tables A, B and C. How many tuples does the result of the following SQL query contains?

```
SELECT A.id
FROM   A
WHERE  A.age > ALL (SELECT B.age
                    FROM   B
                    WHERE  B. name = "arun")
```

(A) 4
(B) 3
(C) 0
(D) 1

Answer (B)
The meaning of “ALL” is the A.Age should be greater than all the values returned by the subquery. There is no entry with name “arun” in table B. So the subquery will return NULL. If a subquery returns NULL, then the condition becomes true for all rows of A (See this for details). So all rows of table A are selected.

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

62. Database Management Systems | Set 4

Following Questions have been asked in GATE 2011 exam.

1. Consider a relational table with a single record for each registered student with the following attributes.

1. Registration_Number:< Unique registration number for each registered student
2. UID: Unique Identity number, unique at the national level for each citizen
3. BankAccount_Number: Unique account number at the bank. A student can have multiple accounts or joint accounts. This attributes stores the primary account number
4. Name: Name of the Student
5. Hostel_Room: Room number of the hostel

Which of the following options is INCORRECT?
(A) BankAccount_Number is a candidate key
(B) Registration_Number can be a primary key
(C) UID is a candidate key if all students are from the same country
(D) If S is a superkey such that S ∩ UID is NULL then S ∪ UID is also a superkey

Answer (A)
A Candidate Key value must uniquely identify the corresponding row in table. BankAccount_Number is not a candidate key. As per the question “A student can have multiple accounts or joint accounts. This attributes stores the primary account number”. If two students have a joint account and if the joint account is their primary account, then BankAccount_Number value cannot uniquely identify a row.

2) Consider a relational table r with sufficient number of records, having attributes A1, A2,…, An and let 1 <= p <= n. Two queries Q1 and Q2 are given below.

Q1 : $\pi_{A1...A_n}(\sigma_{A_p=c}(r))$ where c is a const
Q2: $\pi_{A1...A_n}(\sigma_{c_1 \le A_p \le c_2}(r))$ where c_1 and c_2 are constants

The database can be configured to do ordered indexing on Ap or hashing on Ap. Which of the following statements is TRUE?
(A) Ordered indexing will always outperform hashing for both queries
(B) Hashing will always outperform ordered indexing for both queries
(C) Hashing will outperform ordered indexing on Q1, but not on Q2
(D) Hashing will outperform ordered indexing on Q2, but not on Q1.

Answer (C)
If record are accessed for a particular value from table, hashing will do better. If records are accessed in a range of values, ordered indexing will perform better. See this for more details.

3) Database table by name Loan_Records is given below.

```
Borrower    Bank_Manager    Loan_Amount
```

```
Ramesh      Sunderajan     10000.00
Suresh      Ramgopal       5000.00
Mahesh      Sunderajan     7000.00
```

What is the output of the following SQL query?

```
SELECT Count(*)
FROM  ( (SELECT Borrower, Bank_Manager
       FROM   Loan_Records) AS S
        NATURAL JOIN (SELECT Bank_Manager,
                             Loan_Amount
                      FROM   Loan_Records) AS T );
```

(A) 3
(B) 9
(C) 5
(D) 6

Answer (C)

Following will be contents of temporary table S

```
Borrower    Bank_Manager
--------------------------
 Ramesh      Sunderajan
 Suresh      Ramgqpal
 Mahesh      Sunderjan
```

Following will be contents of temporary table T

```
Bank_Manager   Loan_Amount
---------------------------
Sunderajan      10000.00
Ramgopal        5000.00
Sunderjan       7000.00
```

Following will be the result of natural join of above two tables. The key thing to note is that the natural join happens on column name with same name which is Bank_Manager in the above example. “Sunderjan” appears two times in Bank_Manager column, so their will be four entries with Bank_Manager as “Sunderjan”.

```
Borrower  Bank_Manager   Load_Amount
-----------------------------------
Ramesh    Sunderajan     10000.00
Ramesh    Sunderajan     7000.00
Suresh    Ramgopal       5000.00
Mahesh    Sunderajan     10000.00
Mahesh    Sunderajan     7000.00
```

4) Consider a database table T containing two columns X and Y each of type integer. After the creation of the table, one record (X=1, Y=1) is inserted in the table.

Let MX and My denote the respective maximum values of X and Y among all records in the table at any point in time. Using MX and MY, new records are inserted in the table 128 times with X and Y values being MX+1, 2*MY+1 respectively. It may be noted that each time after the insertion, values of MX and MY change. What will be the output of the following SQL query after the steps mentioned above are carried out?

```
SELECT Y FROM T WHERE X=7;
```

(A) 127
(B) 255
(C) 129
(D) 257

Answer (A)

```
 X    Y
-------
 1    1
 2    3
 3    7
 4    15
 5    31
 6    63
 7   127
 ......
 ......
```

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

63. Database Management Systems | Set 5

Following Questions have been asked in GATE CS 2010 exam.

1) A relational schema for a train reservation database is given below.
Passenger (pid, pname, age)
Reservation (pid, class, tid)

Table: Passenger

pid	pname	age
0	Sachin	65
1	Rahul	66
2	Sourav	67
3	Anil	69

Table : Reservation

pid	class	tid
0	AC	8200
1	AC	8201
2	SC	8201
5	AC	8203
1	SC	8204
3	AC	8202

What pids are returned by the following SQL query for the above instance of the tables?

```
SLECT pid
FROM Reservation ,
WHERE class 'AC' AND
    EXISTS (SELECT *
       FROM Passenger
       WHERE age > 65 AND
       Passenger. pid = Reservation.pid)
```

(A) 1, 0
(B) 1, 2
(C) 1, 3
(S) 1, 5

Answer (C)
When a subquery uses values from outer query, the subquery is called correlated subquery. The correlated subquery is evaluated once for each row processed by the outer query.

The outer query selects 4 entries (with pids as 0, 1, 5, 3) from Reservation table. Out of these selected entries, the subquery returns Non-Null values only for 1 and 3.

2) Which of the following concurrency control protocols ensure both conflict serialzability and freedom from deadlock?
I. 2-phase locking

II. Time-stamp ordering

(A) I only

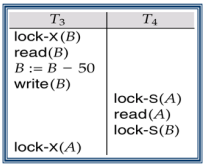
(B) II only

(C) Both I and II

(D) Neither I nor II

Answer (B)

2 Phase Locking (2PL) is a concurrency control method that guarantees serializability. The protocol utilizes locks, applied by a transaction to data, which may block (interpreted as signals to stop) other transactions from accessing the same data during the transaction's life. 2PL may be lead to deadlocks that result from the mutual blocking of two or more transactions. See the following situation, neither T3 nor T4 can make progress.



T_3	T_4
lock-X(B)	
read(B)	
$B := B - 50$	
write(B)	
	lock-S(A)
	read(A)
	lock-S(B)
lock-X(A)	

Timestamp-based concurrency control algorithm is a non-lock concurrency control method. In Timestamp based method, deadlock cannot occur as no transaction ever waits.

3) Consider the following schedule for transactions T1, T2 and T3:

T1	T2	T3
Read (X)		
	Read (Y)	
		Read (Y)
	Write (Y)	
Write (X)		
		Write (X)
	Read (X)	
	Write (X)	

Which one of the schedules below is the correct serialization of the above?

(A)T1→T3→T2

(B)T2→T1→T3

(C)T2→T3→T1

(D)T3→T1→T2

Answer (A)

T1 can complete before T2 and T3 as there is no conflict between Write(X) of T1 and the operations in T2 and T3 which occur before Write(X) of T1 in the above diagram. T3 should can complete before T2 as the Read(Y) of T3 doesn't conflict with Read(Y) of T2. Similarly, Write(X) of T3 doesn't conflict with Read(Y) and Write(Y) operations of T2.

Another way to solve this question is to create a dependency graph and topologically sort the dependency graph. After topologically sorting, we can see the sequence T1, T3, T2.

4) Which of the following functional dependencies hold for relations R(A, B, C) and S(B, D, E):
B → A,
A → C
The relation R contains 200 tuples and the rel ation S contains 100 tuples. What is the
maximum number of tuples possible in the natural join R▷◁S (R natural join S)
(A) 100
(B) 200
(D) 300
(D) 2000

Answer (A)
From the given set of functional dependencies, it can be observed that B is a candidate key of R. So all 200 values of B must be unique in R. There is no functional dependency given for S. To get the maximum number of tuples in output, there can be two possibilities for S.
1) All 100 values of B in S are same and there is an entry in R that matches with this value. In this case, we get 100 tuples in output.
2) All 100 values of B in S are different and these values are present in R also. In this case also, we get 100 tuples.

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

64. Database Management Systems | Set 6

Following questions have been asked in GATE 2009 CS exam.

1) Consider two transactions T1 and T2, and four schedules S1, S2, S3, S4 of T1 and T2 as given below:
T1 = R1[X] W1[X] W1[Y]
T2 = R2[X] R2[Y] W2[Y]
S1 = R1[X] R2[X] R2[Y] W1[X] W1[Y] W2[Y]
S2 = R1[X] R2[X] R2[Y] W1[X] W2[Y] W1[Y]
S3 = R1[X] W1[X] R2[X] W1[Y] R2[Y] W2[Y]
S1 = R1[X] R2[Y]R2[X]W1[X] W1[Y] W2[Y]
Which of the above schedules are conflict-serializable?
(A) S1 and S2
(B) S2 and S3
(C) S3 only

(D) S4 only

Answer (B)
There can be two possible serial schedules T1 T2 and T2 T1. The serial schedule T1 T2 has the following sequence of operations
R1[X] W1[X] W1[Y] R2[X] R2[Y] W2[Y]
And the schedule T2 T1 has the following sequence of operations.
R2[X] R2[Y] W2[Y] R1[X] W1[X] W1[Y]
The Schedule S2 is conflict-equivalent to T2 T1 and S3 is conflict-equivalent to T1 T2.

2) Let R and S be relational schemes such that R={a,b,c} and S={c}. Now consider the following queries on the database:

I. $\pi_{R-S}(r) - \pi_{R-S}(\pi_{R-S}(r) \times s - \pi_{R-S,S}(r))$
II. $\{t \mid t \in \pi_{R-S}(r) \wedge \forall u \in_r (\exists v \in s (u = v[s] \wedge t = v[R-S]))\}$
III. $\{t \mid t \in \pi_{R-S}(r) \wedge \forall v \in r (\exists u \in s (u = v[s] \wedge t = v[R-S]))\}$

```
IV) SELECT R.a, R.b
       FROM R,S
            WHERE R.c=S.c
```

Which of the above queries are equivalent?
(A) I and II
(B) I and III
(C) II and IV
(D) III and IV

Answer (A)
I and II describe the division operator in Relational Algebra and Tuple Relational Calculus respectively. See Page 3 of this and slide numbers 9,10 of this for more details.

3) Consider the following relational schema:

```
Suppliers(sid:integer, sname:string, city:string, street:string)
Parts(pid:integer, pname:string, color:string)
Catalog(sid:integer, pid:integer, cost:real)
```

Consider the following relational query on the above database:

```
SELECT S.sname
    FROM Suppliers S
        WHERE S.sid NOT IN (SELECT C.sid
                            FROM Catalog C
                            WHERE C.pid NOT IN (SELECT P.pid
                                                FROM Parts P
```

```
                                        WHERE P.color<> 'blue'))
```

Assume that relations corresponding to the above schema are not empty. Which one of the following is the correct interpretation of the above query?

(A) Find the names of all suppliers who have supplied a non-blue part.
(B) Find the names of all suppliers who have not supplied a non-blue part.
(C) Find the names of all suppliers who have supplied only blue parts.
(D) Find the names of all suppliers who have not supplied only blue parts.

Answer (A)
The subquery "*SELECT P.pid FROM Parts P WHERE P.color<> 'blue'*" gives pids of parts which are not blue. The bigger subquery "*SELECT C.sid FROM Catalog C WHERE C.pid NOT IN (SELECT P.pid FROM Parts P WHERE P.color<> 'blue')*" gives sids of all those suppliers who have supplied blue parts. The complete query gives the names of all suppliers who have supplied a non-blue part

4) Assume that, in the suppliers relation above, each supplier and each street within a city has a unique name, and (sname, city) forms a candidate key. No other functional dependencies are implied other than those implied by primary and candidate keys. Which one of the following is TRUE about the above schema?
(A) The schema is in BCNF
(B) The schema is in 3NF but not in BCNF
(C) The schema is in 2NF but not in 3NF
(D) The schema is not in 2NF

Answer (A)
A relation is in BCNF if for every one of its dependencies X ? Y, at least one of the following conditions hold:

```
X ? Y is a trivial functional dependency (Y ? X)
X is a superkey for schema R
```

Since (sname, city) forms a candidate key, there is no non-tirvial dependency X ? Y where X is not a superkey

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

65. Computer Networks | Set 1

Following questions have been asked in GATE CS 2012 exam.

1) The protocol data unit(PDU) for the application layer in the Internet stack is
(A) Segment
(B) Datagram
(C) Message
(D) Frame

Answer (C)
The Protocol Data Unit for Application layer in the Internet Stack (or TCP/IP) is called Message.

2) Which of the following transport layer protocolss is used to support electronic mail?
(A) SMTP
(B) IP
(C) TCP
(D) UDP

Answer (C)
E-mail uses SMTP as application layer protocol. SMTP uses TCP as transport layer protocol.

3) In the IPv4 addressing format, the number of networks allowed under Class C addresses is
(A) 2^14
(B) 2^7
(C) 2^21
(D) 2^24

Answer (C)
In class C, 8 bits are reserved for Host Id and 24 bits are reserved for Network Id. Out of these 24 Network Id bits, the leading 3 bits are fixed as 110. So remaining 21 bits can be used for different networks. See this for more details.

4) An Internet Service Provider(ISP) has the following chunk of CIDR-based IP addresses available with it:245.248.128.0/20. The ISP wants to give half of this chunk of addreses to Organization A, and a quarter to Organization B, while retaining the remaining with itself. Which of the following is a valid allocation of addresses to A and B?
(A) 245.248.136.0/21 and 245.248.128.0/22
(B) 245.248.128.0/21 and 245.248.128.0/22
(C) 245.248.132.0/22 and 245.248.132.0/21
(D) 245.248.136.0/22 and 245.248.132.0/21

Answer (A)

Since routing prefix is 20, the ISP has 2^(32-20) or 2^12 addresses. Out of these 2^12 addresses, half (or 2^11) addresses have to be given to organization A and quarter (2^10) addresses have to be given to organization B. So routing prefix for organization A will be 21. For B, it will be 22. If we see all options given in question, only options (A) and (B) are left as only these options have same number of routing prefixes. Now we need to choose from option (A) and (B).
To assign addresses to organization A, ISP needs to take first 20 bits from 245.248.128.0 and fix the 21st bit as 0 or 1. Similarly, ISP needs to fix 21st and 22nd bits for organization B. If we take a closer look at the options (A) and (B), we can see the 21st and 22nd bits for organization B are considered as 0 in both options. So 21st bit of organization A must be 1. Now take the first 20 bits from 245.248.128.0 and 21st bit as 1, we get addresses for organization A as 245.248.136.0/21

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

66. Operating Systems | Set 16

Following questions have been asked in GATE CS 2005 exam.

1) Normally user programs are prevented from handling I/O directly by I/O instructions in them. For CPUs having explicit I/O instructions, such I/O protection is ensured by having the I/O instructions privileged. In a CPU with memory mapped I/O, there is no explicit I/O instruction. Which one of the following is true for a CPU with memory mapped I/O?
(a) I/O protection is ensured by operating system routine(s)
(b) I/O protection is ensured by a hardware trap
(c) I/O protection is ensured during system configuration
(d) I/O protection is not possible

Answwer (a)
Memory mapped I/O means, accessing I/O via general memory access as opposed to specialized IO instructions. An example,

```
unsigned int volatile const *pMappedAddress const = (unsigned int *)0x100;
```

So, the programmer can directly access any memory location directly. To prevent such an access, the OS (kernel) will divide the address space into kernel space and user space. An user application can easily access user application. To access kernel space, we need system calls (traps).
Thanks to Venki for providing the above explanation.

2) What is the swap space in the disk used for?
(a) Saving temporary html pages
(b) Saving process data
(c) Storing the super-block
(d) Storing device drivers

Answer (b)
Swap space is typically used to store process data. See this for more details.

3) Increasing the RAM of a computer typically improves performance because:
(a) Virtual memory increases
(b) Larger RAMs are faster
(c) Fewer page faults occur
(d) Fewer segmentation faults occur

Answer (c)

4) Suppose n processes, P1, …. Pn share m identical resource units, which can be reserved and released one at a time. The maximum resource requirement of process Pi is Si, where Si > 0. Which one of the following is a sufficient condition for ensuring that deadlock does not occur?
(a) $\forall i, s_i < m$ (b) $\forall i, s_i < n$
(c) $\sum_{i=1}^{n} s_i < (m+n)$ (d) $\sum_{i=1}^{n} s_i < (m * n)$

Answer (c)
In the extreme condition, all processes acquire Si-1 resources and need 1 more resource. So following condition must be true to make sure that deadlock never occurs.
$\sum_{i=1}^{n}(Si-1) < m$

The above expression can be written as following.
$\sum_{i=1}^{n} Si < (m + n)$

See this forum thread for an example.

5) Consider the following code fragment:

```
if (fork() == 0)
{ a = a + 5; printf("%d,%d\n", a, &a); }
else { a = a -5; printf("%d, %d\n", a, &a); }
```

Let u, v be the values printed by the parent process, and x, y be the values printed by the child process. Which one of the following is TRUE?
(a) u = x + 10 and v = y
(b) u = x + 10 and v != y

(c) u + 10 = x and v = y
(d) u + 10 = x and v != y

Answer (c)
fork() returns 0 in child process and process ID of child process in parent process.
In Child (x), a = a + 5
In Parent (u), a = a – 5;
Therefore x = u + 10.
The physical addresses of 'a' in parent and child must be different. But our program accesses virtual addresses (assuming we are running on an OS that uses virtual memory). The child process gets an exact copy of parent process and virtual address of 'a' doesn't change in child process. Therefore, we get same addresses in both parent and child. See this run for example.
Thanks to Smart Pointer for providing the above explanation.

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

67. Database Management Systems | Set 7

Following questions have been asked in GATE 2008 CS exam.

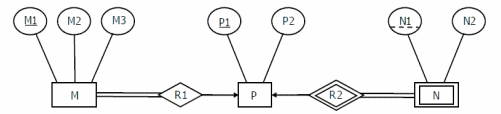
1) Let R and S be two relations with the following schema
R (P,Q,R1,R2,R3)
S (P,Q,S1,S2)
Where {P, Q} is the key for both schemas. Which of the following queries are equivalent?

I. $\Pi_P (R \bowtie S)$
II. $\Pi_P (R) \bowtie \Pi_P (S)$
III. $\Pi_P (\Pi_{P,Q} (R) \cap \Pi_{P,Q} (S))$
IV. $\Pi_P (\Pi_{P,Q} (R) - (\Pi_{P,Q} (R) - \Pi_{P,Q} (S)))$

(A) Only I and II
(B) Only I and III
(C) Only I, II and III
(D) Only I, III and IV

Answer (D)
In I, Ps from natural join of R and S are selected.
In III, all Ps from intersection of (P, Q) pairs present in R and S.
IV is also equivalent to III because (R – (R – S)) = R ∩ S.
II is not equivalent as it may also include Ps where Qs are not same in R and S.

2) Consider the following ER diagram.



The minimum number of tables needed to represent M, N, P, R1, R2 is
(A) 2
(B) 3
(C) 4
(D) 5

Answer (A)
Many-to-one and one-to-many relationship sets that are total on the many-side can be represented by adding an extra attribute to the “many” side, containing the primary key of the “one” side. Since R1 is many to one and participation of M is total, M and R1 can be combined to form the table {M1, M2, M3, P1}. N is a week entity set, so it can be combined with P.

3) Which of the following is a correct attribute set for one of the tables for the correct answer to the above question?
(A) {M1, M2, M3, P1}
(B) {M1, P1, N1, N2}
(C) {M1, P1, N1}
(D) {M1, P1}

Answer (A)

4) Consider the following relational schemes for a library database:
Book (Title, Author, Catalog_no, Publisher, Year, Price)
Collection (Title, Author, Catalog_no)

with in the following functional dependencies:

```
I. Title Author --> Catalog_no
II. Catalog_no --> Title Author Publisher Year
III. Publisher Title Year --> Price
```

Assume {Author, Title} is the key for both schemes. Which of the following statements is true?
(A) Both Book and Collection are in BCNF
(B) Both Book and Collection are in 3NF only
(C) Book is in 2NF and Collection is in 3NF
(D) Both Book and Collection are in 2NF only

Answer (C)

Table Collection is in BCNF as there is only one functional dependency “Title Author –> Catalog_no” and {Author, Title} is key for collection. Book is not in BCNF because Catalog_no is not a key and there is a functional dependency “Catalog_no –> Title Author Publisher Year”. Book is not in 3NF because non-prime attributes (Publisher Year) are transitively dependent on key [Title, Author]. Book is in 2NF because every non-prime attribute of the table is either dependent on the key [Title, Author], or on another non prime attribute.

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

68. Database Management Systems | Set 8

Following questions have been asked in GATE 2005 CS exam.

1) Which one of the following statements about normal forms is FALSE?
(a) BCNF is stricter than 3NF
(b) Lossless, dependency-preserving decomposition into 3NF is always possible
(c) Lossless, dependency-preserving decomposition into BCNF is always possible
(d) Any relation with two attributes is in BCNF

Answer (c)
It is not always possible to decompose a table in BCNF and preserve dependencies. For example, a set of functional dependencies {AB –> C, C –> B} cannot be decomposed in BCNF. See this for more details.

2) The following table has two attributes A and C where A is the primary key and C is the foreign key referencing A with on-delete cascade.

```
A   C
-----
2   4
3   4
4   3
5   2
7   2
9   5
6   4
```

The set of all tuples that must be additionally deleted to preserve referential integrity when the tuple (2,4) is deleted is:
(a) (3,4) and (6,4)

(b) (5,2) and (7,2)
(c) (5,2), (7,2) and (9,5)
(d) (3,4), (4,3) and (6,4)

Answer (C)
When (2,4) is deleted. Since C is a foreign key referring A with delete on cascade, all entries with value 2 in C must be deleted. So (5, 2) and (7, 2) are deleted. As a result of this 5 and 7 are deleted from A which causes (9, 5) to be deleted.

3) The relation book (title, price) contains the titles and prices of different books. Assuming that no two books have the same price, what does the following SQL query list?

```
select title
from book as B
where (select count(*)
   from book as T
   where T.price > B.price) < 5
```

(a) Titles of the four most expensive books
(b) Title of the fifth most inexpensive book
(c) Title of the fifth most expensive book
(d) Titles of the five most expensive books

Answer (d)
When a subquery uses values from outer query, the subquery is called correlated subquery. The correlated subquery is evaluated once for each row processed by the outer query.
The outer query selects all titles from book table. For every selected book, the subquery returns count of those books which are more expensive than the selected book. The where clause of outer query will be true for 5 most expensive book. For example count (*) will be 0 for the most expensive book and count(*) will be 1 for second most expensive book.

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

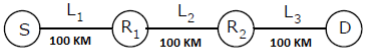
69. Computer Networks | Set 2

Following Questions have been asked in GATE 2012 CS exam.

1) Consider a source computer (S) transmitting a file of size 106 bits to a

destination computer (D) over a network of two routers (R1 and R2) and three links (L1, L2 and L3). L1 connects S to R1;L2 connects R1 to R2; and L3 connects R2 to D. Let each link be of length 100km. Assume signals travel over each link at a speed of 10^8 meters per second. Assume that the link bandwidth on each link is 1Mbps. Let the file be broken down into 1000 packets each of size 1000 bits. Find the total sum of transmission and propagation delays in transmitting the file from S to D?
(A) 1005ms
(B) 1010ms
(C) 3000ms
(D) 3003ms

Answer (A)



Propagation delay to travel from S to R2 = (Distance) / (Link Speed) = 10^5/10^8 = 1ms
Total prorogation delay to travel from S to D = 3*1 ms = 3ms

Total Ttransmission delay for 1 packet = 3 * (Number of Bits) / Bandwidth = 3* (1000/10^6) = 3ms.

The first packet will take 6ms to reach D. While first packet was reaching D, other packets must have been processing in parallel. So D will receive remaining packets 1 packet per 1 ms from R2. So remaining 999 packets will take 999 ms. And total time will be 999 + 6 = 1005 ms

2) Consider an instance of TCP's Additive Increase Multiplicative Decrease(AIMD) algorithm where the window size at the start of the slow start phase is 2 MSS and the threshold at the start of the first transmission is 8 MSS. Assume that a time out occurs during the fifth transmission. Find the congestion window size at the end of the tenth transmission.
(A) 8 MSS
(B) 14 MSS
(C) 7 MSS
(D) 12 MSS

Answer (C)

Since Slow Start is used, window size is increased by the number of segments successfuly sent. This happens until either threshold value is reached or time out occurs. In both of the above situations AIMD is used to avoid congestion. If threshold is reached, window size will be increased linearly. If there is timeout, window size will be reduced to half.

Window size for 1st transmission = 2 MSS

Window size for 2nd transmission = 4 MSS
Window size for 3rd transmission = 8 MSS
threshold reached, increase linearly (according to AIMD)
Window size for 4th transmission = 9 MSS
Window size for 5th transmission = 10 MSS
time out occurs, resend 5th with window size starts with as slow start.
Window size for 6th transmission = 2 MSS
Window size for 7th transmission = 4 MSS
threshold reached, now increase linearly (according to AIMD)
Additive Increase: 5 MSS (since 8 MSS isn't permissible anymore)
Window size for 8th transmission = 5 MSS
Window size for 9th transmission = 6 MSS
Window size for 10th transmission = 7 MSS

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

70. Computer Networks | Set 3

Following Questions have been asked in GATE 2011 CS exam.

1) A layer-4 firewall (a device that can look at all protocol headers up to the transport layer) CANNOT
(A) block HTTP traffic during 9:00PM and 5:00AM
(B) block all ICMP traffic
(C) stop incoming traffic from a specific IP address but allow outgoing traffic to same IP
(D) block TCP traffic from a specific user on a specific IP address on multi-user system during 9:00PM and 5:00AM

Answer (A)
HTTP is an application layer protocol. Since firewal is at layer 4, it cannot block HTTP data.

2) Consider different activities related to email.

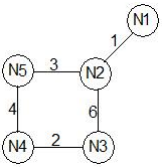
```
m1:Send an email from a mail client to mail server
m2:Download an email from mailbox server to a mail client
m3:Checking email in a web browser
```

Which is the applicable level protocol user in each activity?
(A) m1:HTTP, m2:SMTP, m3:POP
(B) m1:SMTP, m2:FTP, m3:HTTP

(C) m1:SMTP, m2:POP, m3:HTTP
(D) m1:POP, m2:SMTP, m3:IMAP

Answer (C)
Simple Mail Transfer Protocol (SMTP) is typically used by user clients for sending mails.
Post Office Protocol (POP) is used by clients for receiving mails.
Checking mails in web browser is a simple HTTP process.

3) Consider a network with five nodes, N1 to N5, as shown below.



The network uses a a Distance Vector Routing Distance Vector Routing protocol. Once the routes have stabilized, the distance vectors at different nodes are as following.
N1:(0, 1, 7, 8, 4)
N2:(1, 0, 6, 7, 3)
N3:(7, 6, 0, 2, 6)
N4:(8, 7, 2, 0, 4)
N5:(4, 3, 6, 4, 0)
Each distance vector is the distance of the best known path at that instance to nodes, N1 to N5, where the distance to itself is 0. Also, all links are symmetric and the cost is identical in both directions. In each round, all nodes exchange their distance vectors with their respective neighbours. Then all nodes update their distance vectors. In between two rounds, any change in cost of a link will cause the two incident nodes to change only that entry in their distance vectors.

The cost of link N2-N3 reduces to 2 (in both directions). After the next round of update what will be the new distance vector at node, N3?
(A) (3, 2, 0, 2, 5)
(B) (3, 2, 0, 2, 6)
(C) (7, 2, 0, 2, 5)
(D) (7, 2, 0, 2, 6)

Answer (A)
In the next round, every node will send and receive distance vectors to and from neighbors, and update its dostance vector.
N3 will recieve (1, 0, 2, 7, 3) from N2 and it will update distances to N1 and N5 as 3 and 5 respectively.

4) After the update in the previous question, the link N1-N2 goes down. N2 will

reflect this change immediately in its distance vector as cost, ∞. After the NEXT ROUND of update, what will be cost to N1 in the distance vector of N3?
(A) 3
(B) 9
(C) 10
(D) ∞

Answer (C)
In the next round, N3 will receive distance from N2 to N1 as infinite. It will receive distance from N4 to N1 as 8. So it will update distance to N1 as 8 + 2.

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

71. Computer Networks | Set 4

Following Questions have been asked in GATE 2010 CS exam.

1) One of the header fields in an IP datagram is the Time to Live (TTL) field. Which of the following statements best explains the need for this field?
(A) It can be used to priortize packets
(B) It can be used to reduce delays
(C) It can be used to optimize throughput
(D) It can be used to prevent packet looping

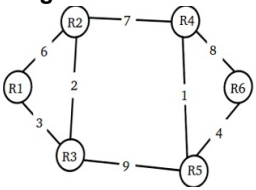
Answer (D)
Time to Live can be thought as an upper bound on the time that an IP datagram can exist in the network. The purpose of the TTL field is to avoid a situation in which an undeliverable datagram keeps circulating.

2) Suppose computers A and B have IP addresses 10.105.1.113 and 10.105.1.91 respectively and they both use the same netmask N. Which of the values of N given below should not be used if A and B should belong to the same network?
(A) 255.255.255.0
(B) 255.255.255.128
(C) 255.255.255.192
(D) 255.255.255.224

(D)
The last octets of IP addresses of A and B are 113 (**011**10001) and 91 (**010**11011). The netmask in option (D) has first three bits set in last octet. If netmask has first 3 bits set, then these bits nmust be same in A and B, but that is not the case. In simple words, we

can say option (D) is not a valid netmask because doing binary '&' of it with addresses of A and B doesn't give the same network address. It must be same address as A and B are on same network. See this for more details.

3) Consider a network with 6 routers R1 to R6 connected with links having weights as shown in the following diagram



All the routers use the distance vector based routing algorithm to update their routing tables. Each router starts with its routing table initialized to contain an entry for each neighbour with the weight of the respective connecting link. After all the routing tables stabilize, how many links in the network will never be used for carrying any data?

(A) 4
(B) 3
(C) 2
(D) 1

Answer (C)
We can check one by one all shortest distances. When we check for all shortest distances for Ri we don't need to check its distances to R0 to Ri-1 because the network graph is undirected.
Following will be distance vectors of all nodes.

Shortest Distances from R1 to R2, R3, R4, R5 and R6
R1 (5, 3, 12, 12, 16)
Links used: R1-R3, R3-R2, R2-R4, R3-R5, R5-R6

Shortest Distances from R2 to R3, R4, R5 and R6
R2 (2, 7, 8, 12)
Links used: R2-R3, R2-R4, R4-R5, R5-R6

Shortest Distances from R3 to R4, R5 and R6
R3 (9, 9, 13)
Links used: R3-R2, R2-R4, R3-R5, R5-R6

Shortest Distances from R4 to R5 and R6
R4 (1, 5)
Links used: R4-R5, R5-R6

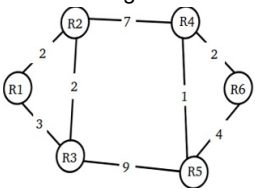
Shortest Distance from R5 to R6
R5 (4)

Links Used: R5-R6

If we mark, all the used links one by one, we can see that following links are never used.
R1-R2
R4-R6

4) Suppose the weights of all unused links in the previous question are changed to 2 and the distance vector algorithm is used again until all routing tables stabilize. How many links will now remain unused?
(A) 0
(B) 1
(C) 2
(D) 3

Answer (B)
After the weights of unused links () are changed to following graph.



Following will be distance vectors of all nodes
R1 (2, 3, 9, 10, 11)
Links used: R1-R2, R1-R3, R2-R4, R4-R5, R4-R6

R2 (2, 7, 8, 9)
Links used: R2-R3, R2-R4, R4-R5, R4-R6

R3 (9, 9, 11)
Links used: R3-R2, R2-R4, R3-R5, R4-R6

R4 (1, 2)
Links used: R4-R5, R4-R6

R5 (3)
Links Used: R5-R4, R4-R6

If we mark, all the used links one by one, we can see that all links are used except the following link.
R5-R6

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

72. Computer Networks | Set 5

Following questions have been asked in GATE CS 2005 exam.

1) Packets of the same session may be routed through different paths in:
(a) TCP, but not UDP
(b) TCP and UDP
(c) UDP, but not TCP
(d) Neither TCP nor UDP

Answer (b)
Packet is the Network layer Protocol Data Unit (PDU). TCP and UDP are Transport layer protocols. Packets of same session may be routed through different routes. Most networks don't use static routing, but use some form of adaptive routing where the paths used to route two packets for same session may be different due to congestion on some link, or some other reason.

2) The address resolution protocol (ARP) is used for:
(a) Finding the IP address from the DNS
(b) Finding the IP address of the default gateway
(c) Finding the IP address that corresponds to a MAC address
(d) Finding the MAC address that corresponds to an IP address

Answer (d)
Address Resolution Protocol (ARP) is a request and reply protocol used to find MAC address from IP address.

3) The maximum window size for data transmission using the selective reject protocol with n-bit frame sequence numbers is:
(a) 2^n
(b) 2^(n-1)
(c) 2^n – 1
(d) 2^(n-2)

Answer (b)
In Selective Reject (or Selective Repeat), maximum size of window must be half of the maximum sequence number.

4) In a network of LANs connected by bridges, packets are sent from one LAN to another through intermediate bridges. Since more than one path may exist between two LANs, packets may have to be routed through multiple bridges. Why is the spanning tree algorithm used for bridge-routing?

(a) For shortest path routing between LANs
(b) For avoiding loops in the routing paths
(c) For fault tolerance
(d) For minimizing collisions

Answer (b)
The main idea for using Spanning Trees is to avoid loops. See Spanning Tree Protocol for more details.

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above

73. Computer Networks | Set 6

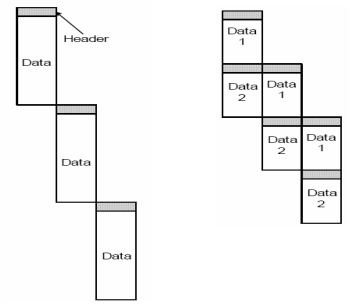
Following questions have been asked in GATE CS 2005 exam.

1) An organization has a class B network and wishes to form subnets for 64 departments. The subnet mask would be:
(a) 255.255.0.0
(b) 255.255.64.0
(c) 255.255.128.0
(d) 255.255.252.0

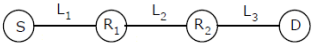
Answer (d)
The size of network ID is 16 bit in class B networks. So bits after 16th bit must be used to create 64 departments. Total 6 bits are needed to identify 64 different departments. Therefore, subnet mask will be 255.255.252.0.

2) In a packet switching network, packets are routed from source to destination along a single path having two intermediate nodes. If the message size is 24 bytes and each packet contains a header of 3 bytes, then the optimum packet size is:
(a) 4
(b) 6
(c) 7
(d) 9

Answer (d)
Dividing a message into packets may decrease the transmission time due to parallelism as shown in the following figure.



But after a certain limit reducing the packet size may increase the transmission time also.
Following figure shows the situation given in question.



Let transmission time to transfer 1 byte for all nodes be t. The first packet will take time = (packet size)*3*t. After the first packet reaches the destination, remaining packets will take time equal to (packet size)*t due to parallelism.

If we use 4 bytes as packet size, there will be 24 packets
Total Transmission time = Time taken by first packet + Time taken by remaining packets = 3*4*t + 23*t = 35t

If we use 6 bytes as packet size, there will be 8 packets
Total Transmission time = 3*6*t + 7*6*t = 50t

If we use 7 bytes as packet size, there will be 6 packets
Total Transmission time = 3*7*t + 5*7*t = 56t

If we use 9 bytes as packet size, there will be 4 packets
Total Transmission time = 3*9*t + 3*9*t = 54t

3) Suppose the round trip propagation delay for a 10 Mbps Ethernet having 48-bit jamming signal is 46.4 ms. The minimum frame size is:
(a) 94
(b) 416
(c) 464
(d) 512

Answer (c)
Transmission Speed = 10Mbps.
Round trip propagation delay = 46.4 ms
The minimum frame size = (Round Trip Propagation Delay) * (Transmission Speed) = 10*(10^6)*46.4*(10^-3) = 464 * 10^3 = 464 Kbit

The concept behind the above formula is collision detection. Consider a situation where a node A wants to send a frame to another node B. When Node A begins transmitting, the signal must propagate the network length. In the worst-case collision scenario, Node B begins to transmit just before the signal for Node A's frame reaches it. The collision signal of Node A and Node B's frame must travel back to Node A for Node A to detect that a collision has occurred.
The time it takes for a signal to propagate from one end of the network to the other is known as the propagation delay. In this worst-case collision scenario, the time that it takes for Node A to detect that its frame has been collided with is twice the propagation delay. Node A's frame must travel all the way to Node B, and then the collision signal must travel all the way from Node B back to Node A. This time is known as the slot time. An Ethernet node must be transmitting a frame for the slot time for a collision with that frame to be detected. This is the reason for the minimum Ethernet frame size.

Source: Microsoft® Windows® Server 2003 TCP/IP Protocols and Services Technical By Joseph Davies

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above

74. Computer Networks | Set 7

Following questions have been asked in GATE CS 2008 exam.

1) Which of the following system calls results in the sending of SYN packets?
(A) socket
(B) bind
(C) listen
(D) connect

Answer (D)
socket() creates a new socket of a certain socket type, identified by an integer number, and allocates system resources to it.
bind() is typically used on the server side, and associates a socket with a socket address structure, i.e. a specified local port number and IP address.
listen() is used on the server side, and causes a bound TCP socket to enter listening state.
connect() is used on the client side, and assigns a free local port number to a socket. In case of a TCP socket, it causes an attempt to establish a new TCP connection.

When connect() is called by client, following three way handshake happens to establish the connection in TCP.

1) The client requests a connection by sending a SYN (synchronize) message to the server.
2) The server acknowledges this request by sending SYN-ACK back to the client.
3) The client responds with an ACK, and the connection is established.

Sources: Berkeley sockets, TCP Connection Establishment and Termination

2) In the slow start phase of the TCP congestion control algorithm, the size of the congestion window
(A) does not increase
(B) increases linearly
(C) increases quadratically
(D) increases exponentially

Answer (D)
Although the name is slow start, during the slow start phase, window size is increased by the number of segments acknowledged, which means window size grows exponentially. This happens until either an acknowledgment is not received for some segment or a predetermined threshold value is reached. See this for more details.

3) If a class B network on the Internet has a subnet mask of 255.255.248.0, what is the maximum number of hosts per subnet?
(A) 1022
(B) 1023
(C) 2046
(D) 2047

Answer (C)
The binary representation of subnet mask is 11111111.11111111.11111000.00000000. There are 21 bits set in subnet. So 11 (32-21) bits are left for host ids. Total possible values of host ids is 2^11 = 2048. Out of these 2048 values, 2 addresses are reserved. The address with all bits as 1 is reserved as broadcast address and address with all host id bits as 0 is used as network address of subnet.
In general, the number of addresses usable for addressing specific hosts in each network is always 2^N – 2 where N is the number of bits for host id.

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above

75. Computer Networks | Set 8

Following questions have been asked in GATE CS 2008 exam.

1) What is the maximum size of data that the application layer can pass on to the TCP layer below?
(A) Any size
(B) 2^16 bytes-size of TCP header
(C) 2^16 bytes
(D) 1500 bytes

Answer (A)
Application layer can send any size of data. There is no limit defined by standards. The lower layers divides the data if needed.

2) A client process P needs to make a TCP connection to a server process S. Consider the following situation: the server process S executes a socket(), a bind() and a listen() system call in that order, following which it is preempted. Subsequently, the client process P executes a socket() system call followed by connect() system call to connect to the server process S. The server process has not executed any accept() system call. Which one of the following events could take place?
(A) connect () system call returns successfully
(B) connect () system call blocks
(C) connect () system call returns an error
(D) connect () system call results in a core dump

Answer (C)
Since accept() call is not executed then connect () gets no response for a time stamp to wait & then return no response server error.

3) A computer on a 10Mbps network is regulated by a token bucket. The token bucket is filled at a rate of 2Mbps. It is initially filled to capacity with 16Megabits. What is the maximum duration for which the computer can transmit at the full 10Mbps?
(A) 1.6 seconds
(B) 2 seconds
(C) 5 seconds
(D) 8 seconds

Answer (B)

```
New tokens are added at the rate of r bytes/sec which is
2Mbps in the given question.

Capacity of the token bucket (b) = 16 Mbits
Maximum possible transmission rate (M) = 10Mbps
So the maximum burst time = b/(M-r) = 16/(10-2) = 2 seconds
```

In the above formula, r is subtracted from M to calculate the maximum burst time. The reason for this subtraction is, new tokens are added at the rate of r while transmission happens at maximum transmission rate M.

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above

76. Computer Networks | Set 9

Following questions have been asked in GATE CS 2007 exam.

1) Which one of the following uses UDP as the transport protocol?
(A) HTTP
(B) Telnet
(C) DNS
(D) SMTP

Answer (C)
DNS primarily uses User Datagram Protocol (UDP) on port number 53 to serve requests. DNS queries consist of a single UDP request from the client followed by a single UDP reply from the server.

2) In Ethernet when Manchester encoding is used, the bit rate is:
(A) Half the baud rate.
(B) Twice the baud rate.
(C) Same as the baud rate.
(D) none of the above

Answer (A)
In Manchester encoding, the bitrate is half of the baud rate.

3) There are n stations in a slotted LAN. Each station attempts to transmit with a probability p in each time slot. What is the probability that ONLY one station transmits in a given time slot?
(A) (1-p)^(n-1)
(B) np(1-p)^(n-1)
(C) p(1-p)^(n-1)
(D) 1-(1-p)^(n-1)

Answer (B)

The probability that a particular station transmits and no body else transmits = p*(1-p)^(n-1)
The probability that any station can transmit = n*(probability that a particular station transmits) = n*p*(1-p)^(n-1). See this for details.

4) In a token ring network the transmission speed is 10^7 bps and the propagation speed is 200 metres/micro second. The 1-bit delay in this network is equivalent to:
(A) 500 metres of cable.
(B) 200 metres of cable.
(C) 20 metres of cable.
(D) 50 metres of cable.

Answer (C)
Transmission delay for 1 bit t = 1/(10^7) = 0.1 micro seconds.
200 meters can be traveled in 1 micro second. Therefore, in 0.1 micro seconds, 20 meters can be traveled.

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above

77. Computer Networks | Set 10

Following questions have been asked in GATE CS 2007 exam.

1) The address of a class B host is to be split into subnets with a 6-bit subnet number. What is the maximum number of subnets and the maximum number of hosts in each subnet?
(A) 62 subnets and 262142 hosts.
(B) 64 subnets and 262142 hosts.
(C) 62 subnets and 1022 hosts.
(D) 64 subnets and 1024 hosts.

Answer (C)
Maximum number of subnets = 2^6-2 =62.
Note that 2 is subtracted from 2^6. The RFC 950 specification reserves the subnet values consisting of all zeros (see above) and all ones (broadcast), reducing the number of available subnets by two.

Maximum number of hosts is 2^10-2 = 1022.
2 is subtracted for Number of hosts is also. The address with all bits as 1 is reserved as broadcast address and address with all host id bits as 0 is used as network address of subnet.

In general, the number of addresses usable for addressing specific hosts in each network is always 2^N – 2 where N is the number of bits for host id.
See this for details

2) The message 11001001 is to be transmitted using the CRC polynomial x^3 + 1 to protect it from errors. The message that should be transmitted is:
(A) 11001001000
(B) 11001001011
(C) 11001010
(D) 110010010011

Answer (B)
The polynomial x^3+1 corresponds to divisor is 1001.

```
11001001 000  <--- input right padded by 3 bits
1001          <--- divisor
01011001 000  <---- XOR of the above 2
 1001         <--- divisor
00010001 000
   1001
00000011 000
      10 01
00000001 010
       1 001
00000000 011 <------- remainder (3 bits)
```

See this for division process.
After dividing the given message 11001001 by 1001, we get the remainder as 011 which is the CRC. The transmitted data is, message + CRC which is 11001001 011.

3) The distance between two stations M and N is L kilometers. All frames are K bits long. The propagation delay per kilometer is t seconds. Let R bits/second be the channel capacity. Assuming that processing delay is negligible, the minimum number of bits for the sequence number field in a frame for maximum utilization, when the sliding window protocol is used, is:

(A) $\left\lceil \log_2 \frac{2LtR + 2K}{K} \right\rceil$
(B) $\left\lceil \log_2 \frac{2LtR}{K} \right\rceil$
(C) $\left\lceil \log_2 \frac{2LtR + K}{K} \right\rceil$
(D) $\left\lceil \log_2 \frac{2LtR + K}{2K} \right\rceil$

Answer (C)

```
Distance between stations = L KM
Propogation delay per KM = t seconds
Total propagation delay = Lt seconds
```

```
Frame size = k bits
Channel capacity = R bits/second
Transmission Time = k/R

Let n be the window size.

UtiliZation = n/(1+2a) where a = Propagation time / transmission time
            = n/[1 + 2LtR/k]
            = nk/(2LtR+k)
For maximum utilization: nk = 2LtR + k
Therefore, n = (2LtR+k)/k
Number of bits needed for n frames is Logn.
```

See this for details.

4) Match the following:

```
(P) SMTP     (1) Application layer
(Q) BGP      (2) Transport layer
(R) TCP      (3) Data link layer
(S) PPP      (4) Network layer
             (5) Physical layer
```

(A) P – 2 Q – 1 R – 3 S – 5
(B) P – 1 Q – 4 R – 2 S – 3
(C) P – 1 Q – 4 R – 2 S – 5
(D) P – 2 Q – 4 R – 1 S – 3

Answer (B)
SMTP is an application layer protocol used for e-mail transmission.
TCP is a core transport layer protocol.
BGP is a network layer protocol backing the core routing decisions on the Internet
PPP is a data link layer protocol commonly used in establishing a direct connection between two networking nodes.

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above

78. Computer Networks | Set 11

Following questions have been asked in GATE CS 2006 exam.

1) Station A uses 32 byte packets to transmit messages to Station B using a sliding window protocol. The round trip delay between A and B is 80 milliseconds and the bottleneck bandwidth on the path between A and B is 128 kbps. What is the optimal window size that A should use?
(A) 20
(B) 40
(C) 160
(D) 320

Answer (B)

```
Round Trip propagation delay = 80ms
Frame size = 32*8 bits
Bandwidth = 128kbps
Transmission Time = 32*8/(128) ms = 2 ms

Let n be the window size.

UtiliZation = n/(1+2a) where a = Propagation time / transmission time
            = n/(1+80/2)

For maximum utilization: n = 41 which is close to option (B)
See this for details.
```

2) Two computers C1 and C2 are configured as follows. C1 has IP address 203.197.2.53 and netmask 255.255.128.0. C2 has IP address 203.197.75.201 and netmask 255.255.192.0. which one of the following statements is true?
(A) C1 and C2 both assume they are on the same network
(B) C2 assumes C1 is on same network, but C1 assumes C2 is on a different network
(C) C1 assumes C2 is on same network, but C2 assumes C1 is on a different network
(D) C1 and C2 both assume they are on different networks.

Answer (C)

```
Network Id of C1 = bitwise '&' of IP of C1 and subnet mask of C1
                 = (203.197.2.53) & (255.255.128.0)
                 = 203.197.0.0
C1 sees network ID of C2 as bitwise '&' of IP of C2 and subnet mask of C1
                = (203.197.75.201) & (255.255.128.0)
                = 203.197.0.0
which is same as Network Id of C1.

Network Id of C2 = bitwise '&' of IP of C2 and subnet mask of C2
                 = (203.197.75.201) & (255.255.192.0)
```

```
                = 203.197.64.0
C2 sees network ID of C1 as bitwise '&' of IP of C1 and subnet mask of C2
                = (203.197.2.53) & (255.255.192.0)
                = 203.197.0.0
which is different from Network Id of C2.
```

Therefore, C1 assumes C2 is on same network, but C2 assumes C1 is on a different network.

3) Station A needs to send a message consisting of 9 packets to Station B using a sliding window (window size 3) and go-back-n error control strategy. All packets are ready and immediately available for transmission. If every 5th packet that A transmits gets lost (but no acks from B ever get lost), then what is the number of packets that A will transmit for sending the message to B?
(A) 12
(B) 14
(C) 16
(D) 18

Answer (C)
Total 16 packets are sent. See following table for sequence of events. Since go-back-n error control strategy is used, all packets after a lost packet are sent again.

```
Sender      Receiver
 1
 2            1
 3            2
 4            3
 5            4
 6
 7            6
              7
          [Timeout for 5]

 5
 6            5
 7            6
 8
 9
              8
              9
          [Timeout for 7]

 7
 8            7
```

```
9             8



         [Timeout for 9]
9
              9
```

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above

79. Computer Networks | Set 12

Following questions have been asked in GATE CS 2009 exam.

1) Let G(x) be the generator polynomial used for CRC checking. What is the condition that should be satisfied by G(x) to detect odd number of bits in error?
(A) G(x) contains more than two terms
(B) G(x) does not divide 1+x^k, for any k not exceeding the frame length
(C) 1+x is a factor of G(x)
(D) G(x) has an odd number of terms.

Answer (C)
Odd number of bit errors can be detected if G(x) contains (x+1) as a factor. See this for proof.

2) Frames of 1000 bits are sent over a 10^6 bps duplex link between two hosts. The propagation time is 25ms. Frames are to be transmitted into this link to maximally pack them in transit (within the link).
What is the minimum number of bits (i) that will be required to represent the sequence numbers distinctly? Assume that no time gap needs to be given between transmission of two frames.
(A) i=2
(B) i=3
(C) i=4
(D) i=5

Answer (D)
Transmission delay for 1 frame = 1000/(10^6) = 1 ms
Propagation time = 25 ms
The sender can atmost transfer 25 frames before the first frame reaches the

destination.
The number of bits needed for representing 25 different frames = 5

3) Consider the data of previous question. Suppose that the sliding window protocol is used with the sender window size of 2^i where is the number of bits identified in the previous question and acknowledgments are always piggybacked. After sending 2^i frames, what is the minimum time the sender will have to wait before starting transmission of the next frame? (Identify the closest choice ignoring the frame processing time.)
(A) 16ms
(B) 18ms
(C) 20ms
(D) 22ms

Answer (B)
Size of sliding window = 2^5 = 32
Transmission time for a frame = 1ms
Total time taken for 32 frames = 32ms
The sender cannot receive acknoledgement before round trip time which is 50ms
After sending 32 frames, the minimum time the sender will have to wait before starting transmission of the next frame = 50 – 32 = 18

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above

80. Database Management Systems | Set 9

Following questions have been asked in GATE 2006 CS exam.

1) Consider the following log sequence of two transactions on a bank account, with initial balance 12000, that transfer 2000 to a mortgage payment and then apply a 5% interest.

```
1. T1 start
2. T1 B old=12000 new=10000
3. T1 M old=0 new=2000
4. T1 commit
5. T2 start
6. T2 B old=10000 new=10500
7. T2 commit
```

Suppose the database system crashes just before log record 7 is written. When the system is restarted, which one statement is true of the recovery procedure?

(A) We must redo log record 6 to set B to 10500
(B) We must undo log record 6 to set B to 10000 and then redo log records 2 and 3
(C) We need not redo log records 2 and 3 because transaction T1 has committed
(D) We can apply redo and undo operations in arbitrary order because they are idempotent.

Answer (C)
Once a transaction is committed, no need to redo or undo operations.

2) Consider the relation enrolled (student, course) in which (student, course) is the primary key, and the relation paid (student, amount) where student is the primary key. Assume no null values and no foreign keys or integrity constraints. Given the following four queries:

```
Query1: select student from enrolled where student in (select student from paid)
Query2: select student from paid where student in (select student from enrolled)
Query3: select E.student from enrolled E, paid P where E.student = P.student
Query4: select student from paid where exists
        (select * from enrolled where enrolled.student = paid.student)
```

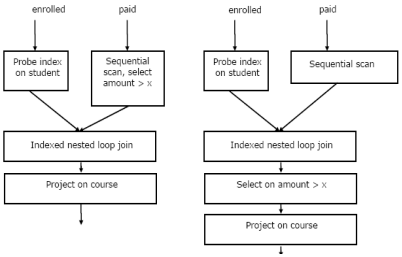
Which one of the following statements is correct?
(A) All queries return identical row sets for any database
(B) Query2 and Query4 return identical row sets for all databases but there exist databases for which Query1 and Query2 return different row sets.
(C) There exist databases for which Query3 returns strictly fewer rows than Query2.
(D) There exist databases for which Query4 will encounter an integrity violation at runtime.

Answer (A)
The output of Query2, Query3 and Query4 will be identical. Query1 may produce duplicate rows. But **rowset** produced by all of them will be same.

```
Table enrolled
student   course
----------------
 abc      c1
 xyz      c1
 abc      c2
 pqr      c1

Table paid
student  amount
```

```
-----------------
 abc      20000
 xyz      10000
 rst      10000


Output of Query 1
 abc
 abc
 xyz

Output of Query 2
 abc
 xyz

Output of Query 3
 abc
 xyz

Output of Query 4
 abc
 xyz
```

3) Consider the relation enrolled(student, course) in which (student, course) is the primary key, and the relation paid(student, amount), where student is the primary key. Assume no null values and no foreign keys or integrity constraints. Assume that amounts 6000, 7000, 8000, 9000 and 10000 were each paid by 20% of the students. Consider these query plans (Plan 1 on left, Plan 2 on right) to "list all courses taken by students who have paid more than x".



A disk seek takes 4ms, disk data transfer bandwidth is 300 MB/s and checking a tuple to see if amount is greater than x takes 10 micro-seconds. Which of the following statements is correct?

(A) Plan 1 and Plan 2 will not output identical row sets for all databases.
(B) A course may be listed more than once in the output of Plan 1 for some databases
(C) For x = 5000, Plan 1 executes faster than Plan 2 for all databases.

(D) For x = 9000, Plan I executes slower than Plan 2 for all databases.

Answer (C)
Assuming that large enough memory is available for all data needed. Both plans need to load both tables courses and enrolled. So disk access time is same for both plans.
Plan 2 does lesser number of comparisons compared to plan 1.
1) Join operation will require more comparisons as the second table will have more rows in plan 2 compared to plan 1.
2) The joined table of two tables will will have more rows, so more comparisons are needed to find amounts greater than x.

4) The following functional dependencies are given:

```
AB   CD, AF   D, DE   F, C  G , F   E, G   A
```

Which one of the following options is false?
(A)CF+ = {ACDEFG}
(B)BG+ = {ABCDG}
(C)AF+ = {ACDEFG}
(D)AB+ = {ABCDFG}

Answer (C)
Closure of AF or AF+ = {ADEF}, closure of AF doesn't contain C and G.
Option (D) also looks correct. AB+ = {ABCDG}, closure of AB doesn't contain F.

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

81. Database Management Systems | Set 10

Following questions have been asked in GATE CS 2005 exam.

1) Let r be a relation instance with schema R = (A, B, C, D). We define r1 = 'select A,B,C from r' and r2 = 'select A, D from r'. Let s = r1 * r2 where * denotes natural join. Given that the decomposition of r into r1 and r2 is lossy, which one of the following is TRUE?
(a) s is subset of r
(b) r U s = r
(c) r is a subset of s
(d) r * s = s

Answer (c)

Consider the following example with lossy decomposition of r into r1 and r2. We can see that r is a subset of s.

```
Table r
 A      B      C      D
----------------------------
 1     10     100    1000
 1     20     200    1000
 1     20     200    1001

Table r1
 A      B      C
------------------
 1     10     100
 1     20     200

Table r2
 A     D
-----------
 1    1000
 1    1001

Table s (natural join of r1 and r2)
 A      B      C      D
----------------------------
 1     10     100    1000
 1     20     200    1000
 1     20     100    1001
 1     20     200    1001
```

2) Let E1 and E2 be two entities in an E/R diagram with simple single-valued attributes. R1 and R2 are two relationships between E1 and E2, where R1 is one-to-many and R2 is many-to-many. R1 and R2 do not have any attributes of their own. What is the minimum number of tables required to represent this situation in the relational model?
(a) 2
(b) 3
(c) 4
(d) 5

Answer (c)
The situation given can be expressed with following sample data.

```
E1
```

```
a
b
c

E2
x
y
z

R1
E1  E2
a    x
a    y
b    z


R2
E1   E2
a     x
a     y
b     y
```

3) Consider a relation scheme R = (A, B, C, D, E, H) on which the following functional dependencies hold: {A–>B, BC–> D, E–>C, D–>A}. What are the candidate keys of R?
(a) AE, BE
(b) AE, BE, DE
(c) AEH, BEH, BCH
(d) AEH, BEH, DEH

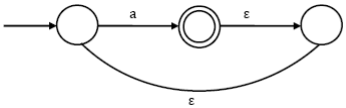
Answer (d)
A set of attributes S is candidate key of relation R if the closure of S is all attributes of R and there is no subset of S whose closure is all attributes of R.
Closure of AEH, i.e. AEH+ = {ABCDEH}
Closure of BEH, i.e. BEH+ = {ABCDEH}
Closure of DEH, i.e. DEH+ = {ABCDEH}

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

82. Automata Theory | Set 2

Following questions have been asked in GATE CS 2012 exam.

1) What is the complement of the language accepted by the NFA shown below? Assume $\sum$ = {a} and ϵ is the empty string



(A) Φ
(B) ϵ
(C) a
(D) {a, ϵ}

Answer (B)
The given alphabet contains only one symbol {a} and the given NFA accepts all strings with any number of occurrences of 'a'. In other words, the NFA accepts a+. Therefore complement of the language accepted by automata is empty string.

2) Given the language L = {ab, aa, baa}, whih of the following strings are in L*?
….1) abaabaaabaa
….2) aaaabaaaa
….3) baaaaabaaaab
….4) baaaaabaa
(A) 1, 2 and 3
(B) 2, 3 and 4
(C) 1, 2 and 4
(D) 1, 3 and 4

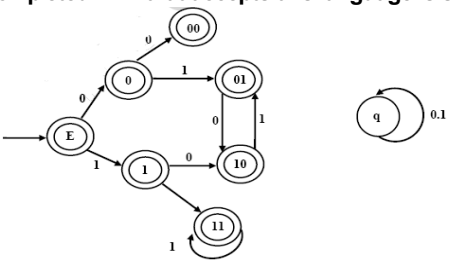
Answer (C)
Any combination of strings in set {ab, aa, baa} will be in L*.
….**1)** "abaabaaabaa" can be partitioned as a combination of strings in set {ab, aa, baa}. The partitions are "ab aa baa ab aa"
….**2)** "aaaabaaaa" can be partitioned as a combination of strings in set {ab, aa, baa}. The partitions are "aa ab aa aa"
….**3)** "baaaaabaaaab" cannot be partitioned as a combination of strings in set {ab, aa, baa}
….**4)** "baaaaabaa" can be partitioned as a combination of strings in set {ab, aa, baa}. The partitions are "baa aa ab aa"

3) Which of the following problems are decidable?
….1) Does a given program ever produce an output?
….2) If L is a context-free language, then is L' (complement of L) also context-free?

….3) If L is a regular language, then is L' also regular?
….4) If L is a recursive language, then, is L' also recursive?
(A) 1, 2, 3, 4
(B) 1, 2,
(C) 2, 3, 4
(D) 3, 4

Answer (D)
….**1)** Is a variation of Turing Machine Halting problem and it is undecidable.
….**2)** Context Free Languages are not closed under intersection and complement. See this for details.
….**3)** Complement of Regular languages is also regular. Then a DFA that accepts the complement of L, i.e. $\sum^*$ – L, can be obtained by swapping its accepting states with its non-accepting states.
….**4)** Recursive Languages are closed under complement. See this for details.

4) Consider the set of strings on {0,1} in which, every substring of 3 symbols has at most two zeros. For examples, 001110 and 011001 are in the language, but 100010 is not. All strings of length less than 3 are also in the language. A partially completed DFA that accepts this language is shown below.



The missing arcs in the DFA are

(A)

	00	01	10	11	q
00	1	0			
01				1	
10	0				
11			0		

(B)

	00	01	10	11	q
00		0			1
01		1			
10				0	
11		0			

(C)

	00	01	10	11	q
00		1			0
01		1			
10			0		
11		0			

(D)

	00	01	10	11	q
00		1			0
01				1	
10	0				
11			0		

Answer (D)
State 'q' is trap state. All other states are accept states. In state 00, DFA must move to 'q' for input symbol 0. All (non-trap) states indicate names indicate the characters seen before reaching that particular state. Option (D) is the only option that follow these rules.

Please write comments if you find any of the answers/explanations incorrect, or you

want to share more information about the topics discussed above.

83. Automata Theory | Set 3

Following questions have been asked in GATE CS 2011 exam.

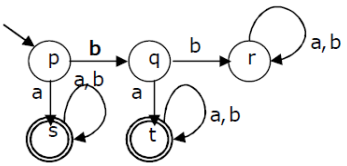
1) The lexical analysis for a modern language such as Java needs the power of which one of the following machine models in a necessary and sufficient sense?
(A) Finite state automata
(B) Deterministic pushdown automata
(C) Non-deterministic pushdown automata
(D) Turing machine

Answer (A)
Lexical analysis is the first step in compilation. In lexical analysis, program is divided into tokens. Lexical analyzers are typically based on finite state automata. Tokens can typically be expressed as different regular expressions:
An identifier is given by [a-zA-Z][a-zA-Z0-9]*
The keyword if is given by if.
Integers are given by [+-]?[0-9]+.

2) Which of the following pairs have DIFFERENT expressive power?
(A) Deterministic finite automata (DFA) and Non-Deterministic finite automata(NFA)
(B) Deterministic push down automata (DPDA) and Non-deterministic pushdown automata
(C) Deterministic single-tape Turing machine and Non-deterministic single-tape Turing Machine
(D) Single-tape Turing machine and multi-tape Turing machine

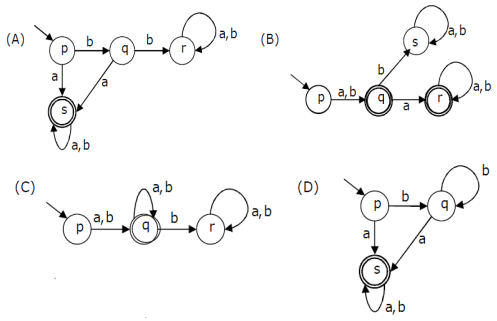
Answer (B)
DPDA cannot handle languages or grammars with ambiguity, but NDPDA can handle languages with ambiguity and any context-free grammar.

3) A deterministic finite automation (DFA)D with alphabet $\sum$ = {a,b} is given below



Which of the following finite state machines is a valid minimal DFA which accepts

the same language as D?



Answer (A)
Options (B) and (C) are invalid because they both accept 'b' as a string which is not accepted by give DFA. D is invalid because it accepts bb+a which are not accepted by given DFA.

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

84. Automata Theory | Set 4

Following questions have been asked in GATE CS 2011 exam.

1) Let P be a regular language and Q be context-free language such that Q $\subseteq$ P. (For example, let P be the language represented by the regular expression p*q* and Q be $\{p^n q^n | n \in N\}$). Then which of the following is ALWAYS regular?
(A) P $\cap$ Q
(B) P – Q
(C) $\sum^*$ – P
(D) $\sum^*$ – Q

Answer (C)
The expression $\sum^*$ – P represents complement of P which is a regular language. Complement of Regular languages is also regular. Then a DFA that accepts the complement of L, i.e. $\sum^*$ – L, can be obtained by swapping its accepting states with its non-accepting states.

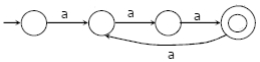
2) Consider the language L1,L2,L3 as given below.
L1={$0^p 1^q$ **| p,q $\in$ N}**

L2={0^p1^q | p,q $\in$ N and p=q}
L3={$0^p1^q1^r$ | p,q,r $\in$ N and p=q=r}
Which of the following statements is NOT TRUE?
(A) Push Down Automata (PDA) can be used to recognize L1 and L2
(B) L1 is a regular language
(C) All the three languages are context free
(D) Turing machine can be used to recognize all the three languages

Answer (C)
The language L3 is not context free. Refer this for more details.

3)Definition of a language L with alphabet {a} is given as following. L= { a^{nk} | k > 0, and n is a positive integer constant} What is the minimum number of states needed in a DFA to recognize L?
(A) k+1
(B) n+1
(C) 2^{n+1}
(D) 2^{k+1}

Answer (B)
Note that n is a constant and k is any positive integer. For example, if n is given as 3, then the DFA must be able to accept 3a, 6a, 9a, 12a, .. To build such a DFA, we need 4 states.



Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

85. Automata Theory | Set 5

Following questions have been asked in GATE CS 2009 exam.

1) S –> aSa| bSb| a| b ;The language generated by the above grammar over the alphabet {a,b} is the set of
(A) All palindromes.
(B) All odd length palindromes.
(C) Strings that begin and end with the same symbol
(D) All even length palindromes.

Answer (B)

The strings accepted by language are {a, b, aaa, bbb, aba, bab, ..}. All of these strings are odd length palindromes.

2) Which one of the following languages over the alphabet {0,1} is described by the regular expression: (0+1)*0(0+1)*0(0+1)*?
(A) The set of all strings containing the substring 00.
(B) The set of all strings containing at most two 0's.
(C) The set of all strings containing at least two 0's.
(D) The set of all strings that begin and end with either 0 or 1.

Answer (C)
The regular expression has two 0's surrounded by (0+1)* which means accepted strings must have at least 2 0's.

3) Which one of the following is FALSE?
(A) There is unique minimal DFA for every regular language
(B) Every NFA can be converted to an equivalent PDA.
(C) Complement of every context-free language is recursive.
(D) Every nondeterministic PDA can be converted to an equivalent deterministic PDA.

Answer (D)
Deterministic PDA cannot handle languages or grammars with ambiguity, but NDPDA can handle languages with ambiguity and any context-free grammar. So every nondeterministic PDA can not be converted to an equivalent deterministic PDA.

4) Match all items in Group 1 with correct options from those given in Group 2.

Group 1	Group 2
P. Regular expression	1. Syntax analysis
Q. Pushdown automata	2. Code generation
R. Dataflow analysis	3. Lexical analysis
S. Register allocation	4. Code optimization

(A) P-4. Q-1, R-2, S-3
(B) P-3, Q-1, R-4, S-2
(C) P-3, Q-4, R-1, S-2
(D) P-2, Q-1, R-4, S-3

Answer (B)

5) . Let L = L1 ∩ L2, where L1 and L2 are languages as defined below:
L1 = {$a^m b^m c a^n b^n$ | m, n >= 0 }
L2 = {$a^i b^j c^k$ | i, j, k >= 0 }

Then L is
(A) Not recursive
(B) Regular
(C) Context free but not regular
(D) Recursively enumerable but not context free.

Answer (C)
The language L1 accept strings {c, abc, abcab, aabbcab, aabbcaabb, …} and L2 accept strings {a, b, c, ab, abc, aabc, aabbc, … }. Intersection of these two languages is L1 ∩ L2 = $\{a^k b^k c \mid k >= 0\}$ which is context free, but not regular.

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

86. Database Management Systems | Set 11

Following questions have been asked in GATE CS 2007 exam.

1) Information about a collection of students is given by the relation *studinfo(studId, name, sex)*. The relation *enroll(studId, courseId)* gives which student has enrolled for (or taken) that course(s). Assume that every course is taken by at least one male and at least one female student. What does the following relational algebra expression represent?

$$\Pi_{\text{courseId}}\left(\left(\Pi_{\text{studId}}\left(\sigma_{sex=\text{"female"}}(\text{studInfo})\right)\times\Pi_{\text{courseId}}(\text{enroll})\right)-\text{enroll}\right)$$

(A) Courses in which all the female students are enrolled.
(B) Courses in which a proper subset of female students are enrolled.
(C) Courses in which only male students are enrolled.
(D) None of the above

Answer (B)
The expression given in question does following steps in sequence.
a) Select *studids* of all female students and selects all *courseids* of all courses.
b) Then the query does a Cartesian Product of the above select two columns from different tables.
c) Finally it subtracts *enroll* table from the result of above step (b). This will remove all the (*studid, courseid*) pairs which are present in *enroll* table. If all female students have registered in a courses, then this course will not be there in the subtracted result.
So the complete expression returns courses in which a proper subset of female students are enrolled.

```
studinfo table
studid   name    sex
```

```
------------------------
 1        a       Male
 2        c       Female
 3        d       Female

enroll table
studid  courseid
------------------
 1         1
 2         1
 3         1
 2         2
 3         3
 3         2


Result of step b
studid     courseid
---------------------
 2             1
 2             2
 2             3
 3             1
 3             2
 3             3


Result of step c
studid    courseid
-------------------
 2           3
```

2) Consider the relation employee(name, sex, supervisorName) with name as the key. supervisorName gives the name of the supervisor of the employee under consideration. What does the following Tuple Relational Calculus query produce?

$$\{e.name \mid employee(e) \wedge (\forall x)[\neg employee(x) \vee x.supervisorName \neq e.name \vee x.sex = \text{"male"}]\}$$

(A) Names of employees with a male supervisor.
(B) Names of employees with no immediate male subordinates.
(C) Names of employees with no immediate female subordinates.
(D) Names of employees with a female supervisor.

Answer (C)
The query selects all those employees whose immediate subordinate is “male”. In other

words, it selects names of employees with no immediate female subordinates

3) Consider the table employee(empId, name, department, salary) and the two queries Q1 ,Q2 below. Assuming that department 5 has more than one employee, and we want to find the employees who get higher salary than anyone in the department 5, which one of the statements is TRUE for any arbitrary employee table?

```
Q1 : Select e.empId
     From employee e
     Where not exists
        (Select * From employee s where s.department = "5" and
                                        s.salary >=e.salary)
Q2 : Select e.empId
     From employee e
     Where e.salary > Any
    (Select distinct salary From employee s Where s.department = "5")
```

(A) Q1 is the correct query
(B) Q2 is the correct query
(C) Both Q1 and Q2 produce the same answer.
(D) Neither Q1 nor Q2 is the correct query

Answer (D)
Consider the following example table.

empid	name	department	salary
1	a	4	90k
2	b	5	30k
3	c	5	50k
4	d	5	80k

Q1 will give empid 1
Q2 will give empid 1, 3, 4
But the correct answer is 4

4) Which one of the following statements if FALSE?
(A) Any relation with two attributes is in BCNF
(B) A relation in which every key has only one attribute is in 2NF
(C) A prime attribute can be transitively dependent on a key in a 3 NF relation.
(D) A prime attribute can be transitively dependent on a key in a BCNF relation.

Answer (D)

5) Consider the following schedules involving two transactions. Which one of the

following statements is TRUE?

$S_1 : r_1(X); r_1(Y); r_2(X); r_2(Y); w_2(Y); w_1(X)$

$S_2 : r_1(X); r_2(X); r_2(Y); w_2(Y); r_1(Y); w_1(X)$

(A) Both S1 and S2 are conflict serializable.
(B) S1 is conflict serializable and S2 is not conflict serializable.
(C) S1 is not conflict serializable and S2 is conflict serializable.
(D) Both S1 and S2 are not conflict serializable.

Answer (C)
S1 is not conflict serializable, but S2 is conflict serializable

```
Schedule S1
   T1             T2
----------------------
  r1(X)
  r1(Y)
                 r2(X)
                 r2(Y)
                 w2(Y)
  w1(X)
The schedule is neither conflict equivalent to T1T2, nor T2T1.

Schedule S2
   T1             T2
----------------------
  r1(X)
                 r2(X)
                 r2(Y)
                 w2(Y)
  r1(Y)
  w1(X)
The schedule is conflict equivalent to T2T1.
```

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above

87. Operating Systems | Set 17

Following question has been asked in GATE 2012 CS exam.

Fetch_And_Add(X,i) is an atomic Read-Modify-Write instruction that reads the

value of memory location X, increments it by the value i, and returns the old value of X. It is used in the pseudocode shown below to implement a busy-wait lock. L is an unsigned integer shared variable initialized to 0. The value of 0 corresponds to lock being available, while any non-zero value corresponds to the lock being not available.

```
AcquireLock(L){
       while (Fetch_And_Add(L,1))
             L = 1;
 }
ReleaseLock(L){
       L = 0;
 }
```

This implementation
(A) fails as L can overflow
(B) fails as L can take on a non-zero value when the lock is actually available
(C) works correctly but may starve some processes
(D) works correctly without starvation

Answer (B)
Take closer look the below while loop.

```
    while (Fetch_And_Add(L,1))
              L = 1;  // A waiting process can be here just after
                      // the lock is released, and can make L = 1.
```

Consider a situation where a process has just released the lock and made L = 0. Let there be one more process waiting for the lock, means executing the AcquireLock() function. Just after the L was made 0, let the waiting processes executed the line L = 1. Now, the lock is available and L = 1. Since L is 1, the waiting process (and any other future coming processes) can not come out of the while loop.

The above problem can be resolved by changing the AcuireLock() to following.

```
AcquireLock(L){
       while (Fetch_And_Add(L,1))
       { // Do Nothing }
 }
```

Source: GATE 2012

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

88. Automata Theory | Set 6

Following questions have been asked in GATE CS 2010 exam.

1) Let L={w $\in$ (0 + 1)*|w has even number of 1s}, i.e. L is the set of all bit strings with even number of 1s. Which one of the regular expression below represents L?
(A) (0*10*1)*
(B) 0*(10*10*)*
(C) 0*(10*1*)*0*
(D) 0*1(10*1)*10*

Answer (B)
Option (A) is incorrect because it cannot accept “110”
Option (C) is incorrect because it accept a string with single 1.
Option (D) is incorrect because it cannot accept 11101

2) Let L1 be a recursive language. Let L2 and L3 be languages that are recursively enumerable but not recursive. Which of the following statements is not necessarily true?
(A) L2 – L1 is recursively enumerable.
(B) L1 – L3 is recursively enumberable
(C) L2 $\cap$ L1 is recursively enumberable
(D) L2 $\cup$ L1 is recursively enumberable

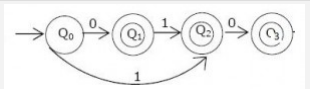
Answer (B)

3) Consider the languages L1={0^i1^j|i != j}, L2={0^i1^j|i = j}, L3 = {0^i1^j|i = 2j+1}, L4 = {0^i1^j |i != 2j}. Which one of the following statements is true?
(A) Only L2 is context free
(B) Only L2 and L3 are context free
(C) Only L1 and L2 are context free
(D) All are context free

Answer (D)
A Pushdown Automata can be built for all four languages.

4) Let w be any string of length n is {0,1}*. Let L be the set of all substrings of w. What is the minimum number of states in a non-deterministic finite automaton that accepts L?
(A) n-1
(B) n
(C) n+1
(D) 2n-1

Answer (C)
We need minimum n+1 states to build NFA that accepts all substrings of a binary string. For example, following NFA accepts all substrings of “010” and it has 4 states.



NFA that accepts all substrings of 010, the substrings are: 0, 01, 10 and 010

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

89. Data Structures and Algorithms | Set 30

Following questions have been asked in GATE CS 2013 exam.

1) Which of the following statements is/are TRUE for an undirected graph?
P: Number of odd degree vertices is even
Q: Sum of degrees of all vertices is even

A) P Only
B) Q Only
C) Both P and Q
D) Neither P nor Q

Answer (C)
Q is true: Since the graph is undirected, every edge increases the sum of degrees by 2.
P is true: If we consider sum of degrees and subtract all even degrees, we get an even number (because Q is true). So total number of odd degree vertices must be even.

2) Consider an undirected random graph of eight vertices. The probability that there is an edge between a pair of vertices is 1/2. What is the expected number of unordered cycles of length three?
(A) 1/8
(B) 1
(C) 7
(D) 8

Answer (C)
A cycle of length 3 can be formed with 3 vertices. There can be total 8C3 ways to pick 3 vertices from 8. The probability that there is an edge between two vertices is 1/2. So expected number of unordered cycles of length 3 = (8C3)*(1/2)^3 = 7

3) What is the time complexity of Bellman-Ford single-source shortest path algorithm on a complete graph of n vertices?
(A) $\Theta(n^2)$
(B) $\Theta(n^2Logn)$
(C) $\Theta(n^3)$
(D) $\Theta(n^3Logn)$

Answer (C).
Time complexity of Bellman-Ford algorithm is $\Theta(VE)$ where V is number of vertices and E is number edges (See this). If the graph is complete, the value of E becomes $\Theta(V^2)$. So overall time complexity becomes $\Theta(V^3)$

4) Which of the following statements are TRUE?
(1) The problem of determining whether there exists a cycle in an undirected graph is in P.
(2) The problem of determining whether there exists a cycle in an undirected graph is in NP.
(3) If a problem A is NP-Complete, there exists a non-deterministic polynomial time algorithm to solve A.

(A) 1,2 and 3
(B) 1 and 2 only
(C) 2 and 3 only
(D) 1 and 3 only

Answer (A)
1 is true because cycle detection can be done in polynomial time using DFS (See this).
2 is true because P is a subset of NP.
3 is true because NP complete is also a subset of NP and NP means **N**on-deterministic **P**olynomial time solution exists. (See this)

5) Which one of the following is the tightest upper bound that represents the time complexity of inserting an object into a binary search tree of n nodes?
(A) O(1)
(B) O(log n)
(C) O(n)
(D) O(n log n)

Answer (C)
The worst case occurs for a skewed tree. In a skewed tree, when a new node is inserted as a child of bottommost node, the time for insertion requires traversal of all node. For example, consider the following tree and the case when something smaller than 70 is inserted.

```
            100
           /
          90
         /
       80
      /
    70
```

6) Which one of the following is the tightest upper bound that represents the number of swaps required to sort n numbers using selection sort?
(A) O(log n)
(B) O(n)
(C) O(n log n)
(D) O(n^2)

Answer (B)
Selection sort requires only O(n) swaps. See this for details.

7) Consider the following operation along with Enqueue and Dequeue operations on
queues, where k is a global parameter

```
MultiDequeue(Q){
   m = k
   while (Q is not empty and m  > 0) {
      Dequeue(Q)
      m = m - 1
   }
}
```

What is the worst case time complexity of a sequence of n MultiDequeue() operations on an initially empty queue?
(A) $\Theta(n)$
(B) $\Theta(n+k)$
(C) $\Theta(nk)$
(D) $\Theta(n^2)$

Answer (A)
Since the queue is empty initially, the condition of while loop never becomes true. So the time complexity is $\Theta(n)$

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

90. Data Structures and Algorithms | Set 31

Following questions have been asked in GATE CS 2013 exam.

1) What is the return value of f(p, p) if the value of p is initialized to 5 before the call? Note that the first parameter is passed by reference, whereas the second parameter is passed by value.

```
int f(int &x, int c) {
   c  = c - 1;
   if (c == 0) return 1;
   x = x + 1;
   return f(x, c) * x;
}
```

(A) 3024
(B) 6561
(C) 55440
(D) 161051

Answer (B)
Since c is passed by value and x is passed by reference, all functions will have same copy of x, but different copies of c.

f(5, 5) = f(x, 4)*x = f(x, 3)*x*x = f(x, 2)*x*x*x = f(x, 1)*x*x*x*x = 1*x*x*x*x = x^4

Since x is incremented in every function call, it becomes 9 after f(x, 2) call. So the value of expression x^4 becomes 9^4 which is 6561.

```
#include <stdio.h>

int f(int &x, int c)
{
    c  = c - 1;
    if (c == 0) return 1;
    x = x + 1;
    return f(x, c) * x;
}
int main()
{
    int p = 5;
    printf("%d", f(p, p));
}
```

1) The preorder traversal sequence of a binary search tree is 30, 20, 10, 15, 25, 23, 39, 35, 42. Which one of the following is the postorder traversal sequence of the

same tree?
(A) 10, 20, 15, 23, 25, 35, 42, 39, 30
(B) 15, 10, 25, 23, 20, 42, 35, 39, 30
(C) 15, 20, 10, 23, 25, 42, 35, 39, 30
(D) 15, 10, 23, 25, 20, 35, 42, 39, 30

Ans (D)
The following is the constructed tree

```
            30
          /      \
        20        39
       /  \      /  \
     10    25  35   42
      \    /
       15 23
```

3) Consider the following function

```
int unknown(int n) {
    int i, j, k = 0;
    for (i  = n/2; i <= n; i++)
        for (j = 2; j <= n; j = j * 2)
            k = k + n/2;
    return k;
}
```

What is the returned value of the above function?
(A) $\Theta(n^2)$
(B) $\Theta(n^2Logn)$
(C) $\Theta(n^3)$
(D) $\Theta(n^3Logn)$

Answer (B)
The outer loop runs n/2 or $\Theta(n)$ times. The inner loop runs $\Theta(Logn)$ times (Note that j is divide by 2 in every iteration). So the statement "k = k + n/2;" runs $\Theta(nLogn)$ times. The statement increases value of k by n/2. So the value of k becomes n/2*$\Theta(nLogn)$ which is $\Theta(n^2Logn)$

4) The number of elements that can be sorted in $\Theta(logn)$ time using heap sort is
(A) $\Theta(1)$
(B) $\Theta(\sqrt{logn})$
(C) $\Theta(Logn/(LogLogn))$
(d) $\Theta(Logn)$

Answer (C)

Time complexity of Heap Sort is $\Theta(mLogm)$ for m input elements. For m = $\Theta(Logn/(LogLogn))$, the value of $\Theta(m * Logm)$ will be $\Theta([Logn/(LogLogn)] * [Log(Logn/(LogLogn))])$ which will be $\Theta([Logn/(LogLogn)] * [LogLogn - LogLogLogn])$ which is $\Theta(Logn)$

5) The procedure given below is required to find and replace certain characters inside an input character string supplied in array A. The characters to be replaced are supplied in array oldc, while their respective replacement characters are supplied in array newc. Array A has a fixed length of five characters, while arrays oldc and newc contain three characters each. However, the procedure is flawed

```
void find_and_replace(char *A, char *oldc, char *newc) {
    for (int i = 0; i < 5; i++)
       for (int j = 0; j < 3; j++)
           if (A[i] == oldc[j]) A[i] = newc[j];
}
```

The procedure is tested with the following four test cases
(1) oldc = "abc", newc = "dab"
(2) oldc = "cde", newc = "bcd"
(3) oldc = "bca", newc = "cda"
(4) oldc = "abc", newc = "bac"
The tester now tests the program on all input strings of length five consisting of characters 'a', 'b', 'c', 'd' and 'e' with duplicates allowed. If the tester carries out this testing with the four test cases given above, how many test cases will be able to capture the flaw?
(A) Only one
(B) Only two
(C) Only three
(D) All four

Answer (B)
The test cases 3 and 4 are the only cases that capture the flaw. The code doesn't work properly when an old character is replaced by a new character and the new character is again replaced by another new character. This doesn't happen in test cases (1) and (2), it happens only in cases (3) and (4).

6) If array A is made to hold the string "abcde", which of the above four test cases will be successful in exposing the flaw in this procedure?
(A) None
(B) 2 only
(C) 3 and 4 only
(D) 4 only

Answer (C)

```c
#include <stdio.h>
#include <string.h>

void find_and_replace(char *A, char *oldc, char *newc) {
    for (int i = 0; i < 5; i++)
       for (int j = 0; j < 3; j++)
           if (A[i] == oldc[j]) A[i] = newc[j];
}

int main()
{
    char *oldc1 = "abc", *newc1 = "dab";
    char *oldc2 = "cde", *newc2 = "bcd";
    char *oldc3 = "bca", *newc3 = "cda";
    char *oldc4 = "abc", *newc4 = "bac";

    char test[] =  "abcde";

    printf("Test 2\n");
    printf("%s\n", test);
    find_and_replace(test, oldc2, newc2);
    printf ("%s\n", test);

    printf("\nTest 3\n");
    strcpy(test, "abcde");
    printf("%s\n", test);
    find_and_replace(test, oldc3, newc3);
    printf ("%s\n", test);

    printf("\nTest 4\n");
    strcpy(test, "abcde");
    printf("%s\n", test);
    find_and_replace(test, oldc4, newc4);
    printf ("%s\n", test);
}
```

Output:

```
Test 2
abcde
abbcd


Test 3
abcde
addde


Test 4
abcde
aacde
```

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

91. Data Structures and Algorithms | Set 32

Following questions have been asked in GATE CS 2014 exam.

1) Let G be a graph with n vertices and m edges. What is the tightest upper bound on the running time on Depth First Search of G? Assume that the graph is represented using adjacency matrix.
(A) O(n)
(B) O(m+n)
(C) $O(n^2)$
(D) O(mn)

Answer: (C)
Explanation: Depth First Search of a graph takes O(m+n) time when the graph is represented using adjacency list.

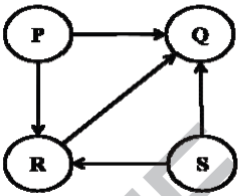
In adjacency matrix representation, graph is represented as an "n x n" matrix. To do DFS, for every vertex, we traverse the row corresponding to that vertex to find all adjacent vertices (In adjacency list representation we traverse only the adjacent vertices of the vertex). Therefore time complexity becomes $O(n^2)$

2) Consider a rooted Binary tree represented using pointers. The best upper bound on the time required to determine the number of subtrees having having exactly 4 nodes $O(n^a Logn^b)$. Then the value of a + 10b is _____

Answer: 1
Explanation: We can find the subtree with 4 nodes in O(n) time. Following can be a simple approach.
1) Traverse the tree in bottom up manner and find size of subtree rooted with current node
2) If size becomes 4, then print the current node.

```
int print4Subtree(struct Node *root)
{
    if (root == NULL)
      return 0;
    int l =  print4Subtree(root->left);
    int r =   print4Subtree(root->right);
    if ((l + r + 1) == 4)
       printf("%d ", root->data);
    return (l + r + 1);
}
```

3) Consider the directed graph given below. Which one of the following is TRUE?



(A) The graph doesn't have any topological ordering
(B) Both PQRS and SRPQ are topological ordering
(C) Both PSRQ and SPRQ are topological ordering
(D) PSRQ is the only topological ordering

Answer: (C)
Explanation: The graph doesn't contain any cycle, so there exist topological ordering.
P and S must appear before R and Q because there are edges from P to R and Q, and from S to R and Q.
See Topological Ordering for more details.

4) Let P be a QuickSort Program to sort numbers in ascending order using the first element as pivot. Let t1 and t2 be the number of comparisons made by P for the inputs {1, 2, 3, 4, 5} and {4, 1, 5, 3, 2} respectively. Which one of the following holds?
(A) t1 = 5
(B) t1 < t2
(C) t1 > t2
(D) t1 = t2

Answer: (C)
Explanation: When first element or last element is chosen as pivot, Quick Sort's worst case occurs for the sorted arrays.
In every step of quick sort, numbers are divided as per the following recurrence.
T(n) = T(n-1) + O(n)

5) Consider the following C function in which size is the number of elements in the array E:
The value returned by the function MyX is the

```
int MyX(int *E, unsigned int size)
{
    int Y = 0;
    int Z;
    int i, j, k;

    for (i = 0; i < size; i++)
        Y = Y + E[i];

    for (i = 0; i < size; i++)
        for (j = i; j < size; j++)
        {
            Z = 0;
            for (k = i; k <= j; k++)
                Z = Z + E[k];
            if (Z > Y)
                Y = Z;
        }
    return Y;
}
```

(A) maximum possible sum of elements in any sub-array of array E.
(B) maximum element in any sub-array of array E.
(C) sum of the maximum elements in all possible sub-arrays of array E
(D) the sum of all the elements in the array E.

Answer: (A)
Explanation: The function does following
Y is used to store maximum sum seen so far and Z is used to store current sum
1) Initialize Y as sum of all elements
2) For every element, calculate sum of all subarrays starting with arr[i]. Store the current sum in Z. If Z is greater than Y, then update Y.

See following for complete solutions of all GATE CS 2014 papers
GATE-CS-2014-(Set-1)
GATE-CS-2014-(Set-2)
GATE-CS-2014-(Set-3)

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

92. C Language | Set 10

Following questions have been asked in GATE CS 2014 exam.

1) Consider the following program in C language:

```c
#include <stdio.h>
main()
{
    int i;
    int *pi = &i;
    scanf("%d", pi);
    printf("%d\n", i+5);
}
```

Which one of the following statements is TRUE?

(A) Compilation fails.

(B) Execution results in a run-time error.

(C) On execution, the value printed is 5 more than the address of variable i.

(D) On execution, the value printed is 5 more than the integer value entered.

Answer: (D)

Explanation: There is no problem in the program as pi points to a valid location. Also, in scanf() we pass address of a variable and pi is an address.

2) Consider the function func shown below:

```c
int func(int num)
{
    int count = 0;
    while (num)
    {
        count++;
        num >>= 1;
    }
    return (count);
}
```

The value returned by func(435)is _____.

Answer: 9

Explanation: The function mainly returns position of Most significant bit in binary representation of n. The MSD in binary representation of 435 is 9th bit.

3) Consider the C function given below.

```c
int f(int j)
{
  static int i = 50;
  int k;
  if (i == j)
  {
    printf("something");
    k = f(i);
    return 0;
  }
  else return 0;
}
```

Which one of the following is TRUE?

(A) The function returns 0 for all values of j.
(B) The function prints the string something for all values of j.
(C) The function returns 0 when j = 50.
(D) The function will exhaust the runtime stack or run into an infinite loop when j = 50

Answer: (D)
Explanation: When j is 50, the function would call itself again and again as neither i nor j is changed inside the recursion.

4) Consider the C function given below. Assume that the array listA contains n (> 0) elements, sorted in ascending order.

```
int ProcessArray(int *listA, int x, int n)
{
    int i, j, k;
    i = 0;
    j = n-1;
    do{
        k = (i+j)/2;
        if (x <= listA[k])
            j = k-1;
        if (listA[k] <= x)
            i = k+1;
    } while (i <= j);
    if (listA[k] == x)
        return(k);
    else
        return -1;
}
```

Which one of the following statements about the function ProcessArray is CORRECT?
(A) It will run into an infinite loop when x is not in listA.
(B) It is an implementation of binary search.
(C) It will always find the maximum element in listA.
(D) It will return -1 even when x is present in listA.

Answer: (B)
Explanation: The program is a simple iterative C implementation of Binary Search.

5) Consider the following function

```
double f(double x)
{
   if (abs(x*x - 3) < 0.01) return x;
   else return f(x/2 + 1.5/x);
}
```

Give a value q (to 2 decimals) such that f(q) will return q:_____.

Answer: 1.732

Explanation:
The main thing to note is the expression “abs(x*x – 3) < 0.01" inside the if condition. The function would return x when x^2 is close to 0 (smaller than 0.01) which means when x is close to square root of 3. Square root of 3 is 1.732.

See following for complete solutions of all GATE CS 2014 papers
GATE-CS-2014-(Set-1)
GATE-CS-2014-(Set-2)
GATE-CS-2014-(Set-3)

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

93. Data Structures and Algorithms | Set 33

Following questions have been asked in GATE CS 2014 exam.

1) Consider the tree arcs of a BFS traversal from a source node W in an unweighted, connected, undirected graph. The tree T formed by the tree arcs is a data structure for computing.
(A) the shortest path between every pair of vertices.
(B) the shortest path from W to every vertex in the graph.
(C) the shortest paths from W to only those nodes that are leaves of T.
(D) the longest path in the graph

Answer: (B)
BFS always produces shortest paths from source to all other vertices in an unweighted graph. The reason is simple, in BFS, we first explore all vertices which are 1 edge away from source, then we explore all vertices which are 2 edges away from the source and so on. This property of BFS makes it useful in many algorithms like Edmonds–Karp algorithm.

2) Consider the following pseudo code. What is the total number of multiplications to be performed?

```
D = 2
for i = 1 to n do
   for j = i to n do
      for k = j + 1 to n do
           D = D * 3
```

(A) Half of the product of the 3 consecutive integers.
(B) One-third of the product of the 3 consecutive integers.
(C) One-sixth of the product of the 3 consecutive integers.
(D) None of the above.
Answer (D)
The statement "D = D * 3" is executed **n*(n+1)*(n-1)/6** times. Let us see how.

For i = 1, the multiplication statement is executed (n-1) + (n-2) + .. 2 + 1 times.
For i = 2, the statement is executed (n-2) + (n-3) + .. 2 + 1 times
……..
……
For i = n-1, the statement is executed once.
For i = n, the statement is not executed at all

So overall the statement is executed following times
[(n-1) + (n-2) + .. 2 + 1] + [(n-2) + (n-3) + .. 2 + 1] + … + 1 + 0

The above series can be written as
S = [n*(n-1)/2 + (n-1)*(n-2)/2 + ….. + 1]

The sum of above series can be obtained by trick of subtraction the series from standard Series S1 = n^2 + $(n-1)^2$ + .. 1^2. The sum of this standard series is n*(n+1)*(2n+1)/6

S1 – 2S = n + (n-1) + … 1 = n*(n+1)/2
2S = n*(n+1)*(2n+1)/6 – n*(n+1)/2
S = n*(n+1)*(n-1)/6

3) Consider a hash table with 9 slots. The hash function is h(k) = k mod 9. The collisions are resolved by chaining. The following 9 keys are inserted in the order: 5, 28, 19, 15, 20, 33, 12, 17, 10. The maximum, minimum, and average chain lengths in the hash table, respectively, are
(A) 3, 0, and 1
(B) 3, 3, and 3
(C) 4, 0, and 1
(D) 3, 0, and 2

Answer: (A)
Following are values of hash function for all keys

```
 5 --> 5
28 --> 1
19 --> 1  [Chained with 28]
15 --> 6
20 --> 2
33 --> 6  [Chained with 15]
```

```
12 --> 3
17 --> 8
10 --> 1 [Chained with 28 and 19]
```

The maximum chain length is 3. The keys 28, 19 and 10 go to same slot 1, and form a chain of length 3.
The minimum chain length 0, there are empty slots (0, 4 and 7).
Average chain length is (0 + 3 + 1 + 1 + 0 + 1 + 2 + 0 + 1)/9 = 1

4) A priority queue is implemented as a Max-Heap. Initially, it has 5 elements. The level-order traversal of the heap is: 10, 8, 5, 3, 2. Two new elements 1 and 7 are inserted into the heap in that order. The level-order traversal of the heap after the insertion of the elements is:
(A) 10, 8, 7, 3, 2, 1, 5
(B) 10, 8, 7, 2, 3, 1, 5
(C) 10, 8, 7, 1, 2, 3, 5
(D) 10, 8, 7, 5, 3, 2, 1

Answer: (A)

```
Initially heap has 10, 8, 5, 3, 2
    10
   /  \
  8    5
 / \
3   2

After insertion of 1
     10
   /   \
  8     5
 / \   /
3   2 1
No need to heapify as 5 is greater than 1.


After insertion of 7
     10
   /   \
  8     5
 / \   / \
3   2 1   7
Heapify 5 as 7 is greater than 5
     10
   /   \
```

```
  8      7
 / \    / \
3   2 1   5
No need to heapify any further as 10 is
greater than 7
```

5) Which one of the following correctly determines the solution of the recurrence relation with T(1) = 1?

```
T(n) = 2T(n/2) + Logn
```

(A) [Tex]\Theta[/Theta](n)
(B) [Tex]\Theta[/Theta](nLogn)
(C) [Tex]\Theta[/Theta](n*n)
(D) [Tex]\Theta[/Theta](log n)

Answer: (A)
This can be solved using Master Method. It falls in case 1.

7) Suppose implementation supports an instruction REVERSE, which reverses the order of elements on the stack, in addition to the PUSH and POP instructions. Which one of the following statements is TRUE with respect to this modified stack?
(A) A queue cannot be implemented using this stack.
(B) A queue can be implemented where ENQUEUE takes a single instruction and DEQUEUE takes a sequence of two instructions.
(C) A queue can be implemented where ENQUEUE takes a sequence of three instructions and DEQUEUE takes a single instruction.
(D) A queue can be implemented where both ENQUEUE and DEQUEUE take a single instruction each.

Answer: (C)
To DEQUEUE an item, simply POP.
To ENQUEUE an item, we can do following 3 operations
1) REVERSE
2) PUSH
3) REVERSE

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above

94. Data Structures and Algorithms | Set 34

Following questions have been asked in GATE CS 2014 exam.

1) Consider the pseudocode given below. The function DoSomething() takes as argument a pointer to the root of an arbitrary tree represented by the leftMostChild-rightSibling representation. Each node of the tree is of type treeNode.

```
typedef struct treeNode* treeptr;
struct treeNode {
    treeptr leftMostChild, rightSibling;
};
int DoSomething (treeptr tree) {
    int value = 0;
    if (tree != NULL) {
        if (tree->leftMostChild == NULL)
            value = 1;
        else
            value = DoSomething(tree->leftMostChild);
        value = value + DoSomething(tree->rightSibling);
    }
    return(value);
}
```

When the pointer to the root of a tree is passed as the argument to DoSomething, the value returned by the function corresponds to the

(A) number of internal nodes in the tree.

(B) height of the tree.

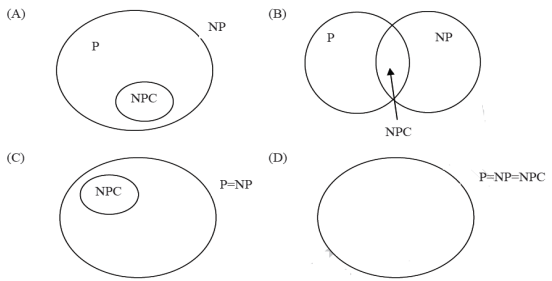
(C) number of nodes without a right sibling in the tree.

(D) number of leaf nodes in the tree.

Answer: (D)

The important thing to note in this question is tree's representation. Tree is represented as leftmost child and right sibling form. So if the leftmost child is NULL for a node, then there is no child of this node. If we take a look at the function, we can notice that the function increments the "value" by 1 only for a leaf node.

2) Suppose a polynomial time algorithm is discovered that correctly computes the largest clique in a given graph. In this scenario, which one of the following represents the correct Venn diagram of the complexity classes P, NP and NP Complete (NPC)?



(A) A
(B) B
(C) C
(D) D

Answer: (D)
Largest Clique is an NP complete problem. If one NP complete problem can be solved in polynomial time, then all of them can be. So NPC set becomes equals to P.

Following are fill in the blanks questions
3) The minimum number of comparisons required to find the minimum and the maximum of 100 numbers is _____.

Answer: 147
The minimum number of comparisons required is 3n/2 – 3 for n numbers. Please refer method 3 of Maximum and minimum of an array using minimum number of comparisons for more details.

4) Consider two strings A = "qpqrr" and B = "pqprqrp". Let x be the length of the longest common subsequence (not necessarily contiguous) between A and B and let y be the number of such longest common subsequences between A and B. Then x + 10y = ___.

Answer: 34
The longest length is 4. There are 3 LCS of length 4 "qprr", "pqrr" and "qpqr".

5) Suppose P, Q, R, S, T are sorted sequences having lengths 20, 24, 30, 35, 50 respectively. They are to be merged into a single sequence by merging together two sequences at a time. The number of comparisons that will be needed in the worst case by the optimal algorithm for doing this is ____.

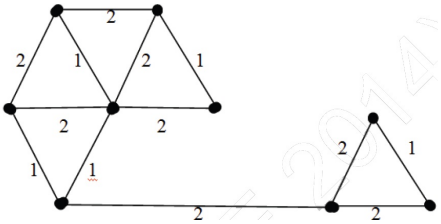
Answer: 358
To merge two lists of size m and n, we need to do m+n-1 comparisons in worst case. Since we need to merge 2 at a time, the optimal strategy would be to take smallest size lists first. The reason for picking smallest two items is to carry minimum items for repetition in merging.
We first merge 20 and 24 and get a list of 44 using 43 worst case comparisons. Then we merge 30 and 35 into a list of 65 using 64 worst case comparisons. Then we merge 50 and 44 into a list of 94 using 93 comparisons. Finally we merge 94 and 65 using 158 comparisons. So total number of comparisons is 43 + 64 + 93 + 158 which is 358.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above

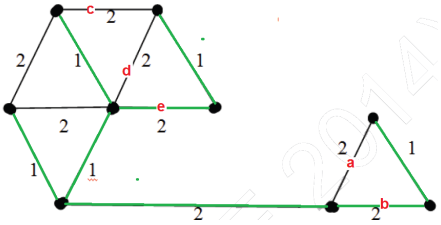
95. Data Structures and Algorithms | Set 35

Following questions have been asked in GATE CS 2014 exam.

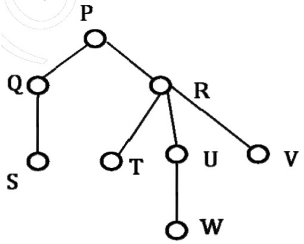
1) The number of distinct minimum spanning trees for the weighted graph below is ____



Answer: 6
Highlighted (in green) are the edges picked to make a MST. In the right side of MST, we could either pick edge 'a' or 'b'. In the left side, we could either pick 'c' or 'd' or 'e' in MST.
There are 2 options for one edge to be picked and 3 options for another edge to be picked. Therefore, total 2*3 possible MSTs.



2) Consider the following rooted tree with the vertex P labeled as root



The order in which the nodes are visited during in-order traversal is
(A) SQPTRWUV
(B) SQPTURWV
(C) SQPTWUVR

(D) SQPTRUWV

Answer: (A)
The only confusion in this question is, there are 3 children of R. So when should R appear – after U or after R? There are two possibilities: SQPTRWUV and SQPTWURV. Only 1st possibility is present as an option A, the IInd possibility is not there. Therefore option A is the right answer.

3) Let A be a square matrix of size n x n. Consider the following program. What is the expected output?

```
C = 100
for i = 1 to n do
    for j = 1 to n do
    {
        Temp = A[i][j] + C
        A[i][j] = A[j][i]
        A[j][i] = Temp - C
    }
for i = 1 to n do
    for j = 1 to n do
        Output(A[i][j]);
```

(A) The matrix A itself
(B) Transpose of matrix A
(C) Adding 100 to the upper diagonal elements and subtracting 100 from diagonal elements of A
(D) None of the above

Answer: A
If we take look at the inner statements of first loops, we can notice that the statements swap A[i][j] and A[j][i] for all i and j.
Since the loop runs for all elements, every element A[l][m] would be swapped twice, once for i = l and j = m and then for i = m and j = l. Swapping twice means the matrix doesn't change.

4) The minimum number of arithmetic operations required to evaluate the polynomial P(X) = $X^5 + 4X^3 + 6X + 5$ for a given value of X using only one temporary variable.
(A) 6
(B) 7
(C) 8
(D) 9

Answer: B
We can parenthesize the polynomial to minimize the number of operations (See Horner's Method). We get $X(X^2(X^2 + 4) + 6) + 5$ after parenthesization.

```
Following is sequence of operations to be used.
Note that we are allowed to use only one variable.
res = X*X
res = res + 4
res = X*res
res = X*res
res = res + 6
res = X*res
res = res + 5
```

5) You have an array of n elements. Suppose you implement quicksort by always choosing the central element of the array as the pivot. Then the tightest upper bound for the worst case performance is

(A) $O(n^2)$
(B) O(nLogn)
(C) Θ(nLogn)
(D) $O(n^3)$

Answer: (A)
The middle element may always be an extreme element (minimum or maximum) in sorted order, therefore time complexity in worst case becomes $O(n^2)$

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above