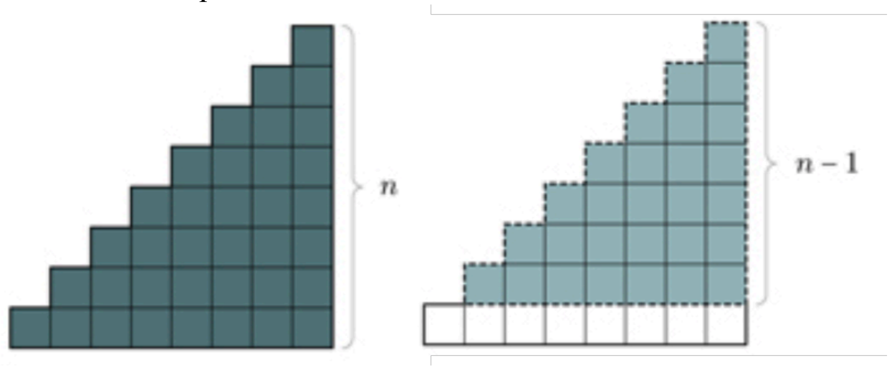
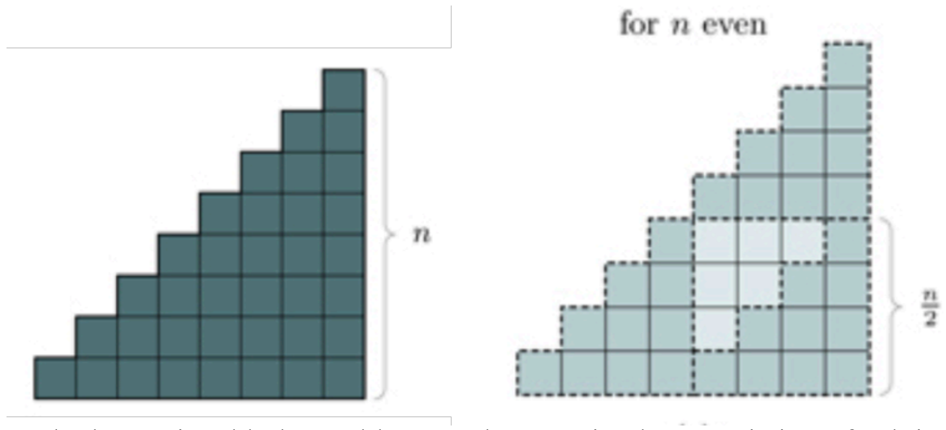
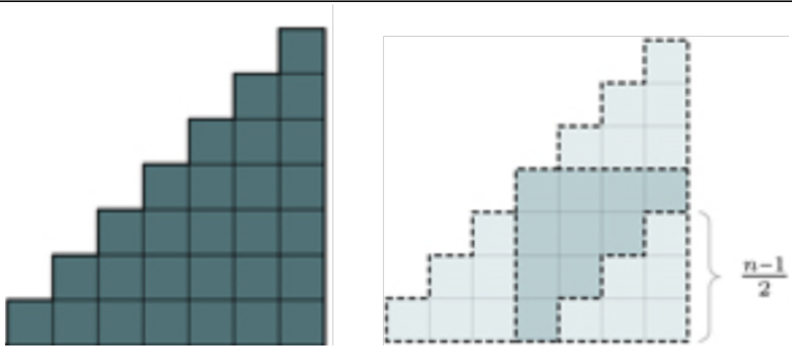
Practice QP ISA1 (2019_20-Even-Sem) for DAA 18CS42

Q No.	Question	Marks
	PART-A	
1(a)	Given an array of n elements, write a recursive algorithm using a sub problem of size n−2 to compute the sum of all elements.	**5**
(b)	A binomial coefficient is recursively defined as ${}^{n}C_{k} = {}^{n-1}C_{k} + {}^{n-1}C_{k-1}$, with base case as ${}^{n}C_{0} = {}^{n}C_{n} = 1$ Compute the value of ${}^{4}C_{3}$ using the above recursion relation, and show the value at each step of iteration	**5**
(c)	Consider Hanoi's tower problem where N discs (numbered from 1 to N) are to be moved from Tower-A to Tower-B using Tower-C with the following constraints: i. Only one disc can be moved in each step. ii. A higher number disc can't be placed on top of another disc. Consider the variant of this Hanoi's tower where you are given an additional tower Tower-D. Thus, the problem now can be stated as move N discs from Tower-A to tower-B using tower-C and tower-D. Write the recurrence relation for this new problem.	**5**
	OR	
2(a)	Consider the problem of finding sum first N integers using the following **Error! Reference source not found.**, where the sum is given by S(N) = n + S(N-1), where shaded area shows the sub problem of size n-1, which is added to N (the number of squares in bottom row.  The same problem can be perceived as solving 4 smaller sub problems as follows, when N is even  And when N is odd, the problem can be perceived as consisting of solving 4 smaller sub problems as	**5**



Thus, perceiving the problem as above, there will be two base cases i.e. when n=1, and when n=2

Define the recurrence relation to solve the above problem of computing sum first N positive integers.

(b) We are trying to determine the worst case time complexity of a library function that is provided to us, whose code we cannot read. We test the function by feeding large numbers of random inputs of different sizes. We find that for inputs of size 300 and 3.000, the function always returns well within one second, but for inputs of size 30,000 it sometimes takes about 1 second and for inputs of size 300,000 it sometimes takes 1-2 minutes. What is a reasonable conclusion we can draw about the worst case time complexity of the library function for the input size N? (You can assume, as usual, that a typical desktop PC performs 10^9 basic operations per second.)

src: NPTel. — **5**

i. (c) Suppose f(n) is $2n^3+4n+5$ and g(n) is $7n^5 + 5n^3 + 12$. Let h(n) be a third, unknown function. Evaluate correctness of each of the following statement

i. h(n) is O(f(n)) and h(n) is also O(g(n))
ii. h(n) is O(f(n)) but h(n) is not O(g(n))
iii. h(n) is O(g(n)) but h(n) is not O(f(n))
iv. h(n) is not O(f(n)) and h(n) is also not O(g(n))

src: NPTel — **5**

PART-B

3(a) How many times is the comparison `i>=n` performed in the following program?

```
int i = 100;
int j = 0;
main(){
  while (i >= j){
    i = i-1;
    j = j+1;
  }
}
```

5

(b) Suppose you are given an array of N integers to sort in ascending order. Evaluate the correctness for each of the following statements:

i. Input in descending order is worst case for selection sort
ii. Input in ascending order is worst case for selection sort
iii. Input in descending order is worst case for insertion sort.
iv. Input in ascending order is worst case for insertion sort.

5

(c) You are given a list of pairs `[("Abhishek",71), ("Shahrukh",85), ("Akshay",71), ("Salman",85), ("Deepika",72), ("Katrina",60), ("Urmila", 65)]`, where each pair consists of a — **5**

	student's name and his/her marks in a course. You are asked sort these pairs in ascending order of marks. What would be the sorted output, if you were told to follow i) Stable Sort ii) Unstable Sort	
	OR	
4(a)	Write an algorithm using iteration to output all prime factors of a 5 given positive integer N.	**5**
(b)	For each of the following situations, evaluate and indicate whether $f=O(g)$, $f=\Omega(g)$, or $f=\Theta(g)$ i. $f=n^{1/2}$, $g=n^{2/3}$ ii. $f=n^2$, $g=5^{(\log_2 n)}$ ii. $f=n^{2n}$, $g=3^n$	**5**
(c)	Show that if c is a positive real number, then $f(n)=1+c+c^2+c^3+\ldots+c^n$ is i. $\Theta(1)$ if $c<1$ ii. $\Theta(n)$ if $c=1$ iii. $\Theta(c^n)$ if $c>1$	**5**