WORKSHEET
PROGRAMM EFFICIENCY

1	What are the Internal Factors on which performance of Algorithm depends?
Ans	1) Time required to run 2) Space(or memory) required to run
2	What are the External Factors on which performance of Algorithm depends?
Ans	1. Size of Input 2. Speed of computer on which it will run 3. Quality of compiler
3	Out of Internal and External factors, which one is the major concern and why?
Ans	Internal factor is major concern for the performance of Algorithm. In Internal factor, Time is the major concern. We can control external factors to sme extent like we can change the computer with high speed computer or we can switch to any other compiler. Even Space factor can also be controlled by purchasing more memory but Time factor we cannot control.
4	What is Time complexity? How time complexity is represented?
Ans	Time complexity mean amount of time taken by an algorithm to run as a function of length of input (n). Time complexity is represented by Big-O notation
5	Algorithm's growth rate is represented by _____
Ans	Big-O
6	The term in a code which decide the efficiency or grown rate of any algorithm is known as _____
Ans	Dominant term
6	**Arrange the following Big-O notations in ascending order of their time taken:** O(N), O(log N), O(N^2), O(2^n), O(1), O(N log N)
Ans	**O(1) < O(log N) < O(N) < O(N log N) < O(N^2)<O(2^n)**

7	**Find out the efficiency of following code using Big-O notation:**
	`n = int(input("Enter n "))`
	`for i in range(n):`
	`  print(i*i)`

Ans	**O(n)**
8	**Consider the following List (L)** L = [10,20,30,15,17,21,42,44,89] *Raj is searching a value 10 & 89 from this list using one by one matching (Linear), What will be the efficiency in term of Big-O when searching:* 10 = _____ 89 = _____
Ans	**10 = O(1)** **89 = O(n)**

9	**Find out the efficiency of following Python code in term of Big-O notation**
	`def print(n):`
	`  for i in range(n):`
	`  for j in range(n):`
	`  print("Row :",i,"Col :",j)`

Ans	**O(n^2)**
10	**Find out the efficiency of following Python code in term of Big-O natation:**

	for i in range(n):
	  if i % 2 == 0:
	  print(i//2)
	  else:
	  print(i**2)

Ans	**O(n)**

11	**Find out the efficiency of following Python code of Binary Search in Big-O notation:**
	def BinarySearch(mylist,item):
	  low=0
	  high=len(mylist)-1
	  while low<=high:
	  mid = (low+high)//2
	  if item == mylist[mid]:
	  return mid+1
	  elif item<mylist[mid]:
	  high=mid-1
	  else:
	  low = mid + 1
	  return -1

Ans	**O(log n)**
12	**Which one is the constant time complexity in term of Big-O notation?** a. O(n) b. O(1) c. O(log n) d. $O(n^2)$
Ans	**O(1)**
13	**Which one is exponential time complexity in term of Big-O notation?** a. O(n) b. O(n log n) c. $O(2^n)$ d. $O(n^2)$
Ans	**$O(2^n)$**
14	**Find the slowest time:** a. O(n) b. $O(n^2)$ c. $O(2^n)$ d. O(n!)
Ans	**O(n!)**
15	**What is the time complexity of Linear Searching?** a. O(n) b. $O(n^2)$ c. O(1) d. $O(2^n)$
Ans	**O(n)**
16	**What is the average time complexity of Insertion Sorting?** a. O(n) b. $O(n^2)$ c. $O(2^n)$ d. O(log n)

Ans	**$O(n^2)$**
17	**Which one is the fastest?** a. O(n) b. O(n log n) c. O(n!) d. O(1)
Ans	**O(1)**
18	**What is the time complexity of following Python code:** a=0 i=n while i>0: a+=I n//=2 a.O(n) b.O(sqrt(n)) c.O(n/2) d.O(log n)
Ans	**O(log n)**
19	**If an algorithm runs in quadratic time, what is time complexity in Big-O natation:** a. O(N) b. O(1) c. $O(N^2)$ d. O(log N)
Ans	**$O(n^2)$**
20	**What is the time complexity of Bubble Sorting?** a. O(n) b. $O(n^2)$ c. O(nlogn) d. O(1)
Ans	**$O(n^2)$**