Q) How do we really Know if n is sufficiently large?

- If Program P actually runs in $10^6 n$ milliseconds while program Q runs in n^2 milliseconds and if we always have $n \leq 10^6$, then, other factors being equal, program Q is faster

log n	n	nlogn	n^2	n^3	2^n
0	1	0	1	1	2
1	2	2	4	8	4
2	4	8	16	64	16
3	8	24	64	512	256
4	16	64	256	4096	65,536
5	32	160	1024	32,768	4,294,967,296

- If a program needs 2^n steps for execution, then when $n=40$, the number of steps needed is approximately 1.1×10^{12}. On a computer performing 1 billion steps per second, this would require about 18.3 minutes. If $n=50$, the same program would run for about 13 days on this computer. When $n=60$, about 310.56 years will be required to execute the program and when $n=100$, about $4*10^{13}$ years will be needed. ∴ For exp programs, utility-$(n \leq 40)$.

- Programs that have a complexity that is a polynomial of high degree are also of limited utility. For ex, if a program needs n^{10} steps, then using our 1 billion steps per second computer we will need 10 seconds when $n=10$; 3171 years when $n=100$; and $3.17*10^{13}$ years when $n=1000$.
- From a practical standpoint, it is evident for reasonably large n (say $n>100$), only Programs of small complexity (such as n, $n\log n$, n^2, n^3) are feasible.

- 4×10^{13} years is about 2900 times the current age of the universe. It is longer than any star will live.

- If $n=10^3$, $2^n = 32*10^{283}$ y on a 1-billion steps per-second computer.

$10^{283} \gg$ all Known physical timescales   Black hole evaporation (supermassive) max 10^{110}.