1. Sort the following functions in increasing order of asymptotic (big-*O*) complexity and explain why:

$f_1(n) = n^{\sqrt{n}}$

$f_2(n) = \log(n^{100000})$

$f_3(n) = 2^{2^{100000}}$

$f_4(n) = 2^n$

$f_5(n) = n^{10} \cdot 2^{n/2}$

$f_6(n) = n\sqrt{n}$

$f_7(n) = \sum_{i=1}^{n} i + 1$

$f_8(n) = 100000n$

2. Solve the following recurrence relations and give a θ bound for each of them.

$(1) T(n) = 9T(n/3) + n$

$$(2)\begin{cases} T(n) = T\left(\frac{n}{2}\right) + T\left(\frac{n}{4}\right) + cn \\ T(1) = 1 \end{cases}$$

$(3) T(n) = T(n-1) + \log n$

3. Divide a positive integer *N* into the sum of several positive integers. How many non-repetitive splitting schemes there are? Give an algorithm with the time complexity $\underline{O(N^2)}$ to do this.

For example, *N* = 5 has seven splitting schemes:

1) **5=5**
2) **5=4+1**
3) **5=3+2**
4) **5=3+1+1**
5) **5=2+2+1**
6) **5=2+1+1+1**
7) **5=1+1+1+1+1**