1.19 求解以下递推方程：

(1) $\begin{cases} T(n)=T(n-1)+n^2 \\ T(1)=1 \end{cases}$

(2) $\begin{cases} T(n)=9T(n/3)+n \\ T(1)=1 \end{cases}$

(3) $\begin{cases} T(n)=T(n/2)+T(n/4)+cn, \quad c\text{为常数} \\ T(1)=1 \end{cases}$

(4) $\begin{cases} T(n)=T(n-1)+\log 3^n \\ T(1)=1 \end{cases}$

(5) $\begin{cases} T(n)=5T(n/2)+(n\log n)^2 \\ T(1)=1 \end{cases}$

(6) $\begin{cases} T(n)=2T(n/2)+n^2\log n \\ T(1)=1 \end{cases}$

(7) $\begin{cases} T(n)=T(n-1)+1/n \\ T(1)=1 \end{cases}$

(8) $T(n)=T(n-1)+\log n$，估计 $T(n)$的阶.

23000129 尹锦润

(1) $T(n)=n^2+(n-1)^2+\cdots+1$

$=\frac{n(n+1)(2n+1)}{6}=\Theta(n^3)$

且$C_1\geq 1$

(3) 假设 $\forall k<n.\ \exists C_1>4C,\ s.t.\ T(k)\leq C_1 k.$

$T(n)=T(n/2)+T(n/4)+cn$

$\leq C_1\frac{n}{2}+C_1\frac{n}{4}+cn$

$\leq C_1 n \quad \therefore T(n)=O(n)$

假设 $\forall k<n.\ s.t.\ T(k)\geq k$

$T(n)=T(n/2)+T(n/4)+cn\geq n$

$\therefore T(n)=\Omega(n)$

$\therefore T(n)=\Theta(n)$

(5) $(n\log n)^2=O(n^{\log_2 5-\varepsilon}) \quad \therefore T(n)=\Theta(n^{\log_2 5})$

(7) $T(n)=T(n-1)+1/n=1+\frac{1}{2}+\cdots+\frac{1}{n}=\ln n+\gamma=\Theta(\log n)$

1.21 设原问题的规模是 n，从下述三个算法中选择一个最坏情况下时间复杂度最低的算法，简要说明理由.

算法 A：将原问题划分规模减半的 5 个子问题，递归求解每个子问题，然后在线性时间将子问题的解合并得到原问题的解.

算法 B：先递归求解 2 个规模为 $n-1$ 的子问题，然后在常量时间内将子问题的解合并.

算法 C：将原问题划分规模为 $n/3$ 的 9 个子问题，递归求解每个子问题，然后在 $O(n^3)$时间将子问题的解合并得到原问题的解.

A: $T(n)=5T(\frac{n}{2})+O(n)=O(n^{\log_2 5})$

B: $T(n)=2T(n-1)+O(1)=O(2^n)$

C: $T(n)=9T(\frac{n}{3})+O(n^3)=O(n^3) \quad \therefore A<C<B$，选 A.

3 Complexity Bounds

For each blank, indicate whether Ai is in O, Ω, or Θ of Bi. More than one space per row can be valid.

A	B	O	Ω	Θ
$10n$	n	✓	✓	✓
10	n	✓	✗	✗
n^2	$2n$	✗	✓	✗
n^{2021}	2^n	✓	✗	✗
$n^{\log 9}$	$9^{\log n}$	✓	✓	✓
$\log(n!)$	$\log(n^n)$	✓	✓	✓
$(3/2)^n$	$(2/3)^n$	✗	✓	✗
3^n	2^n	✗	✓	✗
$n^{1/\log n}$	1	✓	✓	✓
$\log^5 n$	$n^{0.5}$	✓	✗	✗
n^2	$4^{\log n}$	✓	✓	✓
$n^{0.2}$	$(0.2)^n$	✗	✓	✗
$\log\log n$	$\sqrt{\log n}$	✓	✗	✗
$\log(\sqrt{n})$	$\sqrt{\log n}$	✓	✗	✗

4. (a) $O(n^3)$

(b) $$\sum_{i=1}^{n}\sum_{j=i+1}^{n}(j-i+1) = \sum_{i=1}^{n}\frac{(2+n-i+1)(n-i)}{2} = \sum_{i=0}^{n-1}\frac{i(i+3)}{2}$$

$$\geq \sum_{i=0}^{n-1}\frac{i^2}{2} = \frac{(n-1)\cdot n\cdot(2n-1)}{2\times 6}$$

$$= O(n^3)$$

∴ 复杂度也是 $\Omega(n^3)$

(c)

```
For i = 1, 2, ···, n.
Sum ← A[i]
For J = i+1, i+2, ···, n
    Sum ← Sum + A[J]
    B[i, j] ← Sum
Endfor
End for
```

复杂度 $O(n^2)$