二、某算法的计算时间可用下面的递推关系式描述。试采用迭代方式求解该关系式，并用大写О表示解(要求给出详细的推导过程)。(6分)

a n=1，a为常数

T(n)=

2T(n/2) + c\*n n>1，c为常数

|  |  |
| --- | --- |
| **解:**  **首先考虑n是2的整次幂，即对于某个正整数k，有n=2k。**  **T(n) = 2T(n/2) + c\*n**  **= 2(2T(n/4) + c\*n/2) + c\*n**  **= 4T(n/4) + 2c\*n**  **= 4(2T(n/8) + c\*n/4) + 2c\*n**  **= …**  **= 2kT(1) + k\*c\*n**  **= a\*n + c\*n\*log2n**  **所以，T(n)= О(nlogn)。 4分** | **当n不是2的幂时，**  **考虑到T(n)是n的不减函数，**  **即T(2k)≤T(2k+1)，**  **所以，当2k≤n≤2k+1时，**  **有T(n)≤T(2k+1)，所以仍有**  **T(n)=О(nlogn)。**  **2分** |

**二、答案 (过程4分，结果表示2分，共6分)**

二、某算法的计算时间可用下面的递推关系式描述。试采用迭代方式求解该关系式，并用大写О表示解(要求给出详细的推导过程)。(6分)

a *n*=1，a为常数

T(*n*)=

3T(*n*/2) + c\**n* *n*>1，c为常数

（提示：xlogby = ylogbx）

|  |  |
| --- | --- |
| 解:  首先考虑n是2的整次幂，即对于某个正整数k，有n=2k。  T(n) = 3T(n/2) + c\*n  = 3(3T(n/4) + c\*n/2) + c\*n = 32T(n/22) + 3 \*c\*n/2 + c\*n  = 33T(n/23) + (3/2)2 \*c\*n + 3/2 \*c\*n + c\*n  = …  = 3kT(1) +((3/2)k-1 + (3/2)k-2 + … + (3/2)2 + 3/2 + 1)c\*n  =b\*3k +2\*c(3k-2k) ≤ (b+2\*c)\*3k = О(3k) (4分) | 当n不是2的幂时，考虑到T(n)是n的不  减函数，  即有T(2k)≤T(2k+1)，  所以，当2k≤n≤2k+1时，有T(n)≤T(2k+1)，  所以仍有T(n)=О(3k)。  因为，k=log2n，且xlogby = ylogbx  所以，T(n)= О(nlog23)。 （2分） |

二、某算法的计算时间可用下面的递推关系式描述。试采用迭代方式求解该关系式，并用大写О表示解(要求给出详细的推导过程)。(6分)

*b\*d* *n*=1，*b、d*为常数

T(*n*)=

2T(*n*/2) + *d*\**n* *n*>1，*d*为常数

|  |  |
| --- | --- |
| **解：**  当*n*是2的整次幂，即存在某个正整数*k*，使得*n*=2*k*时，有：  T(n)= 2T(*n*/2) + *d* \**n*  = 2(2T(*n*/4) + *d* \**n*/2) + *d* \**n*  = 4T(*n*/4) + 2\* *d* \**n*  = 4(2T(*n*/8) + *d* \**n*/4) + 2 *\*d* \**n*  = …  = 2*k*T(1) + *k*\* *d* \**n*  = b\* *d* \**n* + *d* \**n*\*log2*n*  所以，T(n)= О(*n*log*n*)。 4分 | 当*n*不是2的幂时，考虑到T(*n*)是  *n*的不减函数，所以有  T(2*k*)≤T(2*k*+1)  因此，当2*k*≤*n*≤2*k*+1时，T(*n*)≤T(2*k*+1)，  所以仍有T(*n*)=О(*n*log*n*)。 （2分） |