

DSA Assignment1

罗华坤 软件 02 2019011799

luohk19@mails.tsinghua.edu.cn

1 请写出下列操作的时间复杂度

1.1 $O(1)$

1.2 $O(\log n)$

1.3 $O(n)$

1.4 $O(n^2)$

1.5 $O(n^2)$

1.6 $O(n^3)$ (当两个矩阵都是 $n \times n$ 大小时)

2 最后的结果是 $\{8, 1\}$.

3 请自行举例分析并判断下列结论的正确性

3.1 2^n 的复杂度高于 n 的任意系数的多项式

由于 $\lim_{n \rightarrow \infty} \frac{2^n}{n^c} \stackrel{\text{洛必达}}{=} \lim_{n \rightarrow \infty} \frac{2^n \times (\ln 2)^c}{c!} = \infty \ (c \in R^+)$, 因此该结论是正确的。

下面进行举例分析, 当 n 较小时, 确实存在 2^n 比 n^2 的系数多项式小的情况。但由于指数爆炸现象, 指数函数的增长率远大于多项式的, 而复杂度是对于 n 较大时来说的, 结论正确。

3.2 $\log n$ 的渐进复杂度高于 $O(1)$

同上理, 我们有 $\lim_{n \rightarrow \infty} \frac{c}{\log n} = 0 \ (c \in R^+)$, 因此结论正确。当 $n \rightarrow \infty$, $f(x) = \log_2 n$ 是远大于 $g(x) = 1$ 的

4 请判断下示程序的复杂度, 并说明其功能

该程序的复杂度为 $O(2^n)$, 其功能为当 $n < 0$ 时, 返回 n 本身; 当 $n \geq 0$ 时, 返回第 n 个斐波那契数。

$$\text{即 } func(n) = \begin{cases} n & n < 0 \\ Fib(n) & n \geq 0 \end{cases}$$

5 计算平均查找长度

由于 $ASL = \alpha ASL_{success} + (1 - \alpha) ASL_{fail}$, 而查找成功的平均长度为 $\frac{N+1}{2}$, 失败的为 N

因此 $ASL = \alpha \frac{N+1}{2} + (1 - \alpha)N$

6 $O(n)$

原归并排序的递归式为 $T(n) = 2T(\frac{n}{2}) + O(n)$ ，根据主定理得复杂度为 $O(n \log n)$ 。

若要在最好情况下优化成 $O(n)$ ，即要使得 *Merge* 函数的时间复杂度变为 $O(1)$

(根据主定理 $T(n) = 2T(\frac{n}{2}) + O(1)$ 得到 $T(n) = O(n)$)

具体修改如下：

Algorithm 1: 修改后的 *mergeSort*

```
template<typename T>
void Vector<T>::mergeSort(Rank lo, Rank hi){
    if(hi - lo < 2) return;
    int mi = (lo + hi)/2;
    mergeSort(lo, mi);
    mergeSort(mi, hi);
    if(_elem[mi - 1] > _elem[mi])
        merge(lo, mi, hi);
}
```
