



南京大學

地址：南京市仙林大道163号 邮编：210046
网址：http://www.nju.edu.cn

2012.5.27 有修改

```
public int binary (int a) {  
    if (a % 2 == 1) {  
        return a;  
    }  
    else {  
        while (a % 2 == 1) {  
            a = a / 2;  
        }  
        return a;  
    }  
}  
  
public int recursion (int a) {  
    a = deal(a);  
    if (a == 1) {  
        return 1;  
    }  
    else {  
        return recursion (deal((int)(a/2)) + 1);  
    }  
}
```

```
2. private char[] str, int low, int high) {  
    if (low == high) {  
        System.out.println (str.substring (low, high));  
    }  
    else {  
        for (int i = low; i < high; i++) {  
            swap (str, low, high);  
            permute (str, low + 1, high);  
        }  
    }  
}
```

```

1. public int max(int[] a, int b) {
    if (b == 1) {
        int max;
        max = max(a, b-1);
        if (max > a[b-1]) {
            return max;
        }
        else {
            return a[b-1];
        }
    }
    else {
        return a[b];
    }
}

```

```

2. public float avg(int[] a, int n) {
    if (n == 0) {
        return a[0];
    }
    else {
        return (a[n] + avg(a, n-1)) / n;
    }
}

```

```

4. public int lenail (@Nullable a) {
    if (a == null) {
        return 0;
    }
    else {
        return lenail(a) + 1;
    }
}

```

```

5. public boolean rearrstn (String a) {
    if (a.length() == 1 || a.length() == 2) {
        return true;
    }
    else {
        if (a.charAt(0) == a.charAt(a.length()-1)) {
            return rearrstn(a.substring(1, a.length()-1));
        }
        else {
            return false;
        }
    }
}

```

```

6. public String toU (String a) {
    char[] b = a.toCharArray();
    StringBuilder sb = new StringBuilder();
    for (char c : b) {
        if (Character.isAlphabetic(c)) {
            sb.append(c.toUpperCase());
        }
    }
    return sb.toString();
}

```

1. ~~4+1~~ = 在每行添加一行

$$\sum_{k=1}^{n-1} n-k = nk - \frac{k(k-1)}{2}$$

$$= O(k^2)$$

其化简为 $O(1)$ 的复杂度为 $O(k^2)$

2. c: 每行添加了 $\log N$ 项, 共有 n 项,
故为 $N \log N$, $O(N \log N)$

d: 每行添加了 $\log N$ 项

内层共行 $\frac{n(n-1)}{2}$, 故为 $O(N^2 \log N)$

3. ~~每行添加一行, 共添加一行~~

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

$$\sum_{i=0}^{n-1} \frac{i^2(i^2-1)}{2} = O(N^5)$$

i: 当 $i=0, 1, \dots, i^2-1$ 时 共有三层

三层共添加了 i 次

$$\text{故 共有三层共有 } 0+1+\dots+(i^2-1) \text{ 次, } = \frac{i \times i(i-1)}{2} = \frac{1}{2}i^2 - \frac{1}{2}i$$

共有三层共有 i^2-i 次

$$\text{共 } \frac{1}{2}i^3 + \frac{1}{2}i^2 - i \text{ 次}$$

$$\sum_{i=0}^{n-1} \frac{1}{2}i^3 + \frac{1}{2}i^2 - i, \text{ 故为 } O(N^4)$$

4. (1) 有三层总项 共有 n 项, 1, 2 层共 n^2 项, 共 n^3 项

$$(2) \text{ 内层: } 1+2+\dots+i = \frac{i(i+1)}{2}$$

$$\sum_{i=1}^n \frac{i(i+1)}{2} = \frac{1}{2} \times \frac{n(n+1)(n+2)}{6} + \frac{1}{2} \times \frac{n(n+1)}{2}$$

$$= \frac{2n^3+3n^2+n+3n^2+3n}{12} = \frac{n^3+3n^2+2n}{6}$$

(3) x 从每层 n 次到 1 , 再共行一次 返回操作 $y-1$

$$\text{故共 } 100 \times 11 = 1100 \text{ 项}$$