

# 题解 1

学号: 13350019 姓名: 温志远 提交日期: 2015-11-03

题目链接:

[https://uva.onlinejudge.org/index.php?option=com\\_onlinejudge&Itemid=8&page=show\\_problem&problem=2070](https://uva.onlinejudge.org/index.php?option=com_onlinejudge&Itemid=8&page=show_problem&problem=2070)

题目大意:

输入: 一个数  $n$ , ( $3 \leq n \leq 10000$ );  
输出:  $0, 1, 2, \dots, n-1$  的一种排序, 使得任意长度  
大于等于 3 的子序列都不构成等差序列

思路:

采用递归并且归并的方法, 将序列分为奇数下标和偶数下标, 分出两个序列再递归实现, 直到每个最后的子序列的长度小于 2。

算法步骤:

1. 开一个容量为 10001 的数组存原始序列;
2. 设计一个 `ct` 函数, 输入参数是数组的开始下标和最后一个元素的下标, 输出就是最终的结果序列; 在函数实现过程中通过一个 `temp` 数组存中间值, 实现奇数序列和偶数序列的分离。

算法复杂度:

递归二分数组:  $O(\log(n))$ ;

将每个数组奇偶分离:  $O(n)$ ;

所以最终的算法复杂度为:  $O(n \log(n))$ ;

源代码:

```
# include <iostream>
# include <cstdlib>
using namespace std;
int a[10001], temp[10001];
void show(int n) {
    cout << n << ": ";
    for (int i = 0; i < n; i++) {
        cout << a[i] << " ";
    }
    cout << endl;
}
void ct(int m, int n) {
    if (n > m + 1) {
        for (int i = m; i < n; i++) {
            temp[i] = a[i];
```

```

    }
    for (int i = m, j = m; j < n; i++, j += 2){
        a[i] = temp[j];
    }
    int begin = (m + n) / 2 + (m + n) % 2;
    for (int i = begin, j = m + 1; j < n; i++, j += 2) {
        a[i] = temp[j];
    }
    ct(m, begin);
    ct(begin, n);
}
}
int main() {
    int n;
    while (cin >> n && n != 0) {
        for (int i = 0; i < n; i++) {
            a[i] = i;
        }
        ct(0, n);
        show(n);
    }
}

```

评测结果：

当时在 hust 的 oj 上提交的结果时间和空间都是 0：

4903715	algo13350019	B	Accepted	0	0	C++11 4.8.2	696	2 months ago
---------	--------------	---	----------	---	---	-------------	-----	--------------

然后写题解的时候又直接在 uvaoj 上提交，得到的结果还是 0：

#	Problem	Verdict	Language	Run Time	Submission Date
16627685	11129 An antiarithmetic permutation	Accepted	C++	0.000	2015-12-25 03:42:12

## 题解 2

学号：13350019 姓名：温志远 提交日期：2015-12-07

题目链接：

[https://uva.onlinejudge.org/index.php?option=com\\_onlinejudge&Itemid=8&page=show\\_problem&problem=2829](https://uva.onlinejudge.org/index.php?option=com_onlinejudge&Itemid=8&page=show_problem&problem=2829)

题目大意：

向  $n$  个部下分配  $n$  个任务，给出分配每个任务所需的时间和执行每个任务所需的时间；不能同时给两个部下分配任务，但是所有的部下可以同时执行他们各自的任务；给出最优的分配策略，是任务尽早完成。

输入：

部下一个数  $N$  ( $1 \leq N \leq 1000$ );

N 行两个数: B ( $1 \leq B \leq 10000$ ), J ( $1 \leq J \leq 10000$ ) 是每个任务的交待时间和执行时间

思路:

贪心算法, 由于任务的执行可以并行, 但是任务的分配任务不可以, 所以我们按照 J 的从大到小的顺序给任务排序, 执行时间最短;

算法步骤:

1. 通过一个结构体存每个任务的分配时间和执行时间;
2. 自定义 cmp 函数来排序结构体;
3. 通过一个 maxn 变量来累加每次执行完任务的时间;

算法复杂度:

sort 函数:  $O(\log(n))$ ;

现行写出最大值:  $O(n)$ ;

所以最终的是:  $O(n + \log(n))$ ;  $O(n + \log(n))$

源代码:

```
# include <iostream>
# include <algorithm>
using namespace std;
struct pp {
    int b;
    int j;
};
bool cmp(pp p1, pp p2) {
    return p1.j > p2.j;
}
pp list[1005];

int main(){
    int n, i, ans, maxn, Case = 1;
    while (cin >> n && n != 0) {
        ans = maxn = 0;
        for (i = 0; i < n; i++) {
            cin >> list[i].b >> list[i].j;
        }
        sort(list, list + n, cmp);
        for (i = 0; i < n; i++) {
            ans += list[i].b;
            maxn = max(maxn, ans + list[i].j);
        }
        cout << "Case " << Case << ": " << maxn << endl;
```

```

        Case++;
    }
    return 0;
}

```

评测结果:

5138200	algo13350019	B	Accepted	0	9	C++11 4.8.2	547	18 days ago
---------	--------------	---	----------	---	---	-------------	-----	-------------

## 题解 3

学号: 13350019 姓名: 温志远 提交日期: 2015-11-07

题目链接:

<http://poj.org/problem?id=2533>

题目大意:

给出一序列, 求出该序列的最长递增子序列的最大长度。

思路:

用数组  $a[]$  存储序列,  $b[i]$  表示以  $a[i]$  为结尾的序列的最大长度。因此要求出  $b[i]$  的最大值, 即求出  $\max\{b[0], b[1], \dots, b[i-1]\}$  的最大值, 那么  $b[i]$  的最大  $\max\{b[0], b[1], \dots, b[i-1]\} + 1$ ; 即可写出状态方程:  $b[i] = \max\{b[0], b[1], \dots, b[j]\} + 1; (0 \leq j < i \text{ \& \& } a[j] < a[i])$ , 然后求出数组  $b[]$  中的最大值即为所求。

算法步骤:

1. 初始化  $a, b$  数组;
2. 遍历  $a[]$  数组, 每遇到后者比前者大的, 更新  $b[]$ ;
3. 输出  $b[]$  的最大值。

算法复杂度:

两重 for 循环, 所以是  $O(N^2)$ ;

源代码:

```

#include <iostream>
#include <cmath>
using namespace std;
int a[1002];
int b[1002];
int main() {
    int n;
    cin >> n;
    for (int i = 1; i <= n; i++) {
        cin >> a[i];
        b[i] = 1;
    }
    a[0] = b[0] = 0;
    //b[1] = 1;
}

```

```

for (int i = 1; i <= n; ++i) {
    int temp = 0;
    for (int j = 0; j < i; j++) {
        if (a[i] > a[j])
            temp = max(temp, b[j]);
    }
    b[i] += temp;
}
int maxn = 0 ;
for (int i = 1; i <= n; ++i) {
    maxn = max(maxn, b[i]);
}
cout << maxn << endl;
}

```

评测结果:

4938178( # )	algo13350019	POJ 2533	Accepted	224	16	C++	480	2 months ago
--------------	--------------	----------	----------	-----	----	-----	-----	--------------