

第三次作业 Greedy

姓名：梁付槐

学号：2018Z8013261003

题目一：是否存在无向图

算法：我们设 $D = \{d_1, d_2, \dots, d_n\}$, 每次迭代时, 先对 D 按降序排序, 然后取出第一项, 令 $k = D[1]$, 然后将其移除, 再令 $n = n - 1$ 。然后循环 k 次, 第 j 次循环时另 $D[j]$ 减一并做递归。

伪代码：

```
1  Graph_Judge(D, n) {
2      sort(D);
3      if D[1] == 0
4          return true;
5      if D[n] < 0
6          return false;
7      k = D[1];
8      n = n - 1;
9      delete D[1];
10     for j = 1 to k
11         D[j] = D[j] - 1;
12         Graph_Judge(D, n);
13 }
```

算法的正确性：

如果 D 中最大的数等于 0 而且最小的元素等于 0, 那么 D 中所有的元素均为 0。从这一次迭代的第一步可以看出结果是满足条件的。如果 D 中有值小于 0, 那么前一步的子问题中, 那么 $D[1]$ 就大于点的个数, 这样就不能找到满足条件的图, 所以算法是正确的。

时间复杂度分析： $T(n) = O(n(n \log n + cn)) = O(n^2 \log n)$.

题目二：任务分配

算法：我们按照 f_i 对任务进行降序排序, 然后按照顺序依次在主机上执行, 这样我们就得到了最优解。

伪代码：

```
1  Job_Schedule(J) {  
2      sort(J) //按 $f_i$ 降序排列  
3      ordinal run(J) //依次执行  
4  }  
5
```

算法的正确性：

我们假设 J_1 对应的时间为 p_1 和 f_1 , J_2 对应的时间为 p_2 和 f_2 .

因为分机足够用，所以当任务在主机上执行完毕后可以在分机上立即执行。

所以分四种情况：

1) p_1 在 p_2 前面且 $f_1 > p_2$, 用时: $p_1 + f_1$

2) p_1 在 p_2 前面且 $f_1 < p_2$, 用时: $p_1 + f_1 + f_2$

3) p_1 在 p_2 后面且 $f_1 > p_2$, 用时: $p_2 + p_1 + f_1$

4) p_1 在 p_2 后面且 $f_1 < p_2$, 用时: $p_1 + f_1 + p_2$

不失一般性，我们假设 $f_1 > f_2$, 那么 $time4 - time1 = p_2 > 0$

选择第一种，因此算法是正确的。

时间复杂度分析：

用了一次排序和一次循环，因此 $T(n) = O(n \log n + n) = O(n \log n)$.