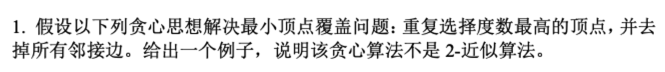
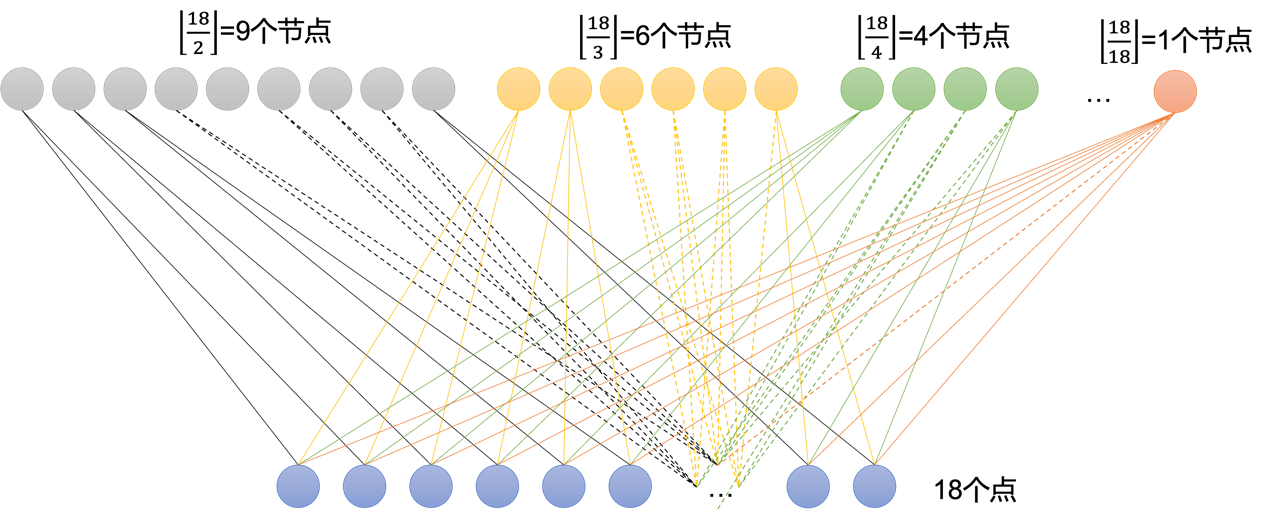
学号：**ZY2206117** 姓名：**黄海浪** 作业：**homework2**

1. **题目1**



如图表1，建立图。下半部分共18个节点，看作，上半部分可以看作有点集合，其中表示该集合中有节点，且表示该集合中每个节点和下半部分的中的个点有边。



图表 1图G

由此，按照题意，中顶点最大度数为17，所以首先选择，然后中最大顶点度数会少1，因此选择，不断计算下去，最终，会选择整个集合，而不是。

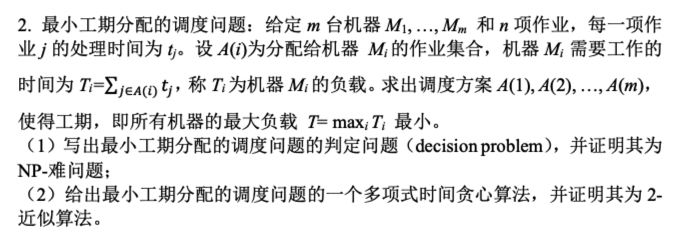
即选取的点为

由图可以很容易知道图的最小顶点覆盖为，即共个点。

所以计算得到：

因此，该贪心算法不是2-近似算法。

1. **题目2**

****

（1）

该问题的判定问题：

是否存在一个调度方案A(1),A(2),…,A(m)，使得最大负载。

证明其为NP-难问题：

1. 证明该问题是NP问题，即多项式时间内可以验证结果。

如果我们找到了一个分配调度方案A(1),A(2),…,A(m)，直接计算所有机器负载，然后与给定负载T进行比较即可，验证的时间复杂度是，即在多项式时间内可以完成。

1. 找到一个NPC问题Q，使得Q≤题目描述问题（最小工期分配调度问题）。

假设Q为partition problem，设其中一个实例为，表示个数的集合，即，给定整数，请问将集合中的数字分到两个集合中去，是否存在一种分配方案使得两个集合中元素的和的差值的绝对值不大于，其中。

根据题目，当最小工期分配调度问题中，及其数量，作业数量，作业时间，则可以构造一个最小工期调度实例。

因为上述两个问题的解都对应一种情况，即问题有解当且仅当问题有解。

综上所述，最小工期分配的调度问题是NP难问题。

（2）

贪心算法：

想法：将所有作业时间按照从大到小排序，依次将作业分配给机器，每次选择负载最小的机器进行分配，直到所有作业分配完毕。

排序的时间复杂度可以计算为，符合多项式时间贪心算法的要求。

证明该算法为2-近似算法：

1. 当时，将所有的作业依次分配给机器，此时解就是最优解，即
2. 当时，考虑最大负载的机器分配的最后一个任务为
   1. 若，即最大负载的机器为，则有
   2. 若，有：

即

因为，所以有，即

综上所述，该贪心算法是2-近似算法。