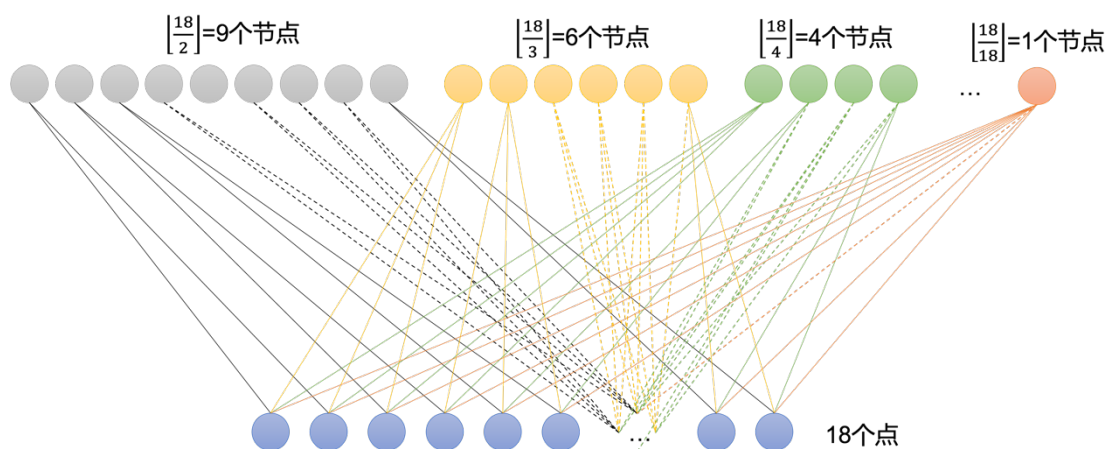


## 1. 题目 1

1. 假设以下列贪心思想解决最小顶点覆盖问题：重复选择度数最高的顶点，并去掉所有邻接边。给出一个例子，说明该贪心算法不是 2-近似算法。

如图表 1，建立图  $G = (V, E)$ 。下半部分共 18 个节点，看作  $V_a$ ，上半部分可以看作有点集合  $\{V_2, V_3, V_4, \dots, V_{18}\}$ ，其中  $V_i$  表示该集合中有  $\lfloor \frac{18}{i} \rfloor$  节点，且表示该集合中每个节点和下半部分的  $V_a$  中的  $i$  个点有边。



图表 1 图 G

由此，按照题意， $V_a$  中顶点最大度数为 17，所以首先选择  $V_{18}$ ，然后  $V_a$  中最大顶点度数会少 1，因此选择  $V_{17}$ ，不断计算下去，最终，会选择  $\{V_2, V_3, V_4, \dots, V_{18}\}$  整个集合，而不是  $V_a$ 。即选取的点为

$$m = \sum_{i=2}^{18} \left\lfloor \frac{18}{i} \right\rfloor = 9 + 6 + 4 + 3 + 3 + 2 + 2 + 2 + 9 \times 1 = 40$$

由图可以很容易知道图  $G$  的最小顶点覆盖为  $A$ ，即共  $m^* = 18$  个点。

所以计算得到：

$$R = \max\left(\frac{m}{m^*}, \frac{m^*}{m}\right) = \frac{40}{18} > 2$$

因此，该贪心算法不是 2-近似算法。

## 2. 题目 2

2. 最小工期分配的调度问题：给定  $m$  台机器  $M_1, \dots, M_m$  和  $n$  项作业，每一项作业  $j$  的处理时间为  $t_j$ 。设  $A(i)$  为分配给机器  $M_i$  的作业集合，机器  $M_i$  需要工作的时间为  $T_i = \sum_{j \in A(i)} t_j$ ，称  $T_i$  为机器  $M_i$  的负载。求出调度方案  $A(1), A(2), \dots, A(m)$ ，

使得工期，即所有机器的最大负载  $T = \max_i T_i$  最小。

(1) 写出最小工期分配的调度问题的判定问题 (decision problem)，并证明其为 NP-难问题；

(2) 给出最小工期分配的调度问题的一个多项式时间贪心算法，并证明其为 2-近似算法。

(1)

该问题的判定问题：

是否存在一个调度方案  $A(1), A(2), \dots, A(m)$ ，使得最大负载  $\max_i T_i \leq T$ 。

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  $A(1), A(2), \dots, A(m)$ ，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  $O(n)$ ，即在多项式时间内可以完成。

2. 找到一个 NPC 问题  $Q$ ，使得  $Q \leq$  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  $I = \{S\}$ ， $S$  表示  $p$  个数的集合，即  $S = \{K_1, K_2, \dots, K_p\}$ ，给定整数  $d$ ，请问将  $S$  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  $d$ ，其中  $d = 0$ 。

根据题目，当最小工期分配调度问题中，及其数量  $m = 2$ ，作业数量  $n = p$ ，作业时间  $t_i = K_i$ ，则可以构造一个最小工期调度实例  $I' = \{m, n, t_1, t_2, \dots, t_n, T\}$ 。

因为上述两个问题的解都对应一种情况，即问题  $I$  有解当且仅当问题  $I'$  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  $O(n^2)$ ，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  $n \leq m$  时，将所有的作业依次分配给机器，此时解就是最优解，即  $T^* = T = \max(t_j)$

2. 当  $n > m$  时，考虑最大负载的机器分配的最后一个任务为  $j^*$

a) 若  $j^* = 1$ ，即最大负载的机器为  $T = \max(t_j)$ ，则有  $T^* = T = \max(t_j)$

b) 若  $j^* \neq 1$ ，有：

$$T - t_{j^*} \leq \frac{1}{m} \sum_{k=1}^m T_k = \frac{1}{m} \sum_{k=1}^{n-\{j^*\}} t_k \leq \frac{1}{m} \sum_{k=1}^n t_k \leq T^*$$

即

$$T \leq T^* + t_{j^*}$$

因为  $t_{j^*} \leq T^*$ ，所以有  $T \leq 2T^*$ ，即

$$\frac{T}{T^*} \leq 2$$

综上所述，该贪心算法是 2-近似算法。