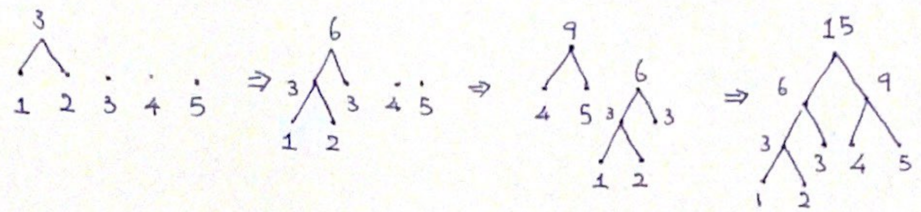
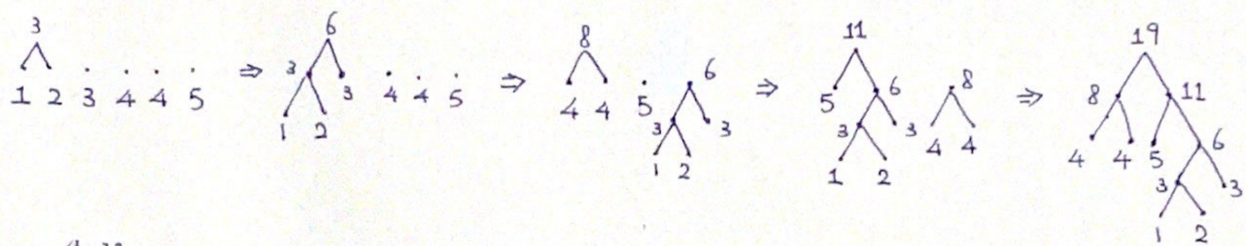
14. (a) 最优二进制编码（含空格）：



最优二进制编码：c: 000，e: 001，s: 01，t: 10，a: 11

原字符串变为：01 10 11 10 001 ␣ 11 000 10 ␣ 1101 ␣ 11 ␣ 01 001 11 10

(b) 最优二进制编码（不含空格）：



编码：c: 1100，e: 1101，s: 111，␣(空格): 00，t: 01，a: 10

原字符串变为：111 01 10 01 1101 00 10 1100 01 00 10 111 00 10 00 111 1101 10 01

1. 首先证：prim 算法可以生成图G的支撑树：

假设第一个加入的结点为 v_1，令集合 $V=\{v_1\}$，此时加入结点 v_2（v_2 不在 V 中），v_2 与 V 中结点相连，边数增加 1，重复以上操作，直至G的所有结点加入 V 中，此时边数恰好为结点数减1，是支撑树。

再证：prim 算法生成最小树

设 T_0 为 prim 算法生成的树，图的最小生成树为 T_1，树的边数为 n.（结点数 $n+1$）

对 T_0 的 n 条边进行排序，由小至大分别为 $e_{01}, e_{02}, \cdots e_{0n}$.

对 T_1 的边从小至大排序，得 $e_{11}, e_{12}, \ldots, e_{1n}$

① 对 T_0, T_1 的边依次比较，若存在 $e_{0k} \in T_0$ 且 $e_{0k} \notin T_1$，其中 $e_{0k}=(V_u, V_e)$

在 T_1 中找到一个以 V_u 为端点的边 $e_{1x} \in T_1$ 且 $e_{1x} \notin T_0$，其中 $e_{1x}=(V_u, V_f)$

e_{0k}, e_{1x} 起点均为 V_u，由 prim 的特性　$W(e_{0k}) \le W(e_{1x})$

故可在 T_1 中删去 e_{1x}，加上 e_{0k}，此时边数不变，顶点数不变，也不存在环

故新树 T_1' 的总权值小于等于 T_1 的总权值，因为 T_1 是最小生成树，故 T_1, T_1' 总权值相等，不断进行操作①，最终得到的树为 T_0，由于每次操作后总权值不变，故 T_0, T_1 的总权值相等，故 T_0 为最小生成树。△