第 9 章 《贪婪技术》习题

1. 为找零问题设计一个贪婪算法，它以金额 n 和硬币的面额$d_1 > d_2 > \cdots > d_m$ 作为输入，给出算法的伪码描述，并分析算法的时间复杂度。

```
coin_change( [d₁,d₂,....dₘ], n )
   num ←0
   for i← n: 1
      while n >= dᵢ
          n ← n - dᵢ
          num ← num + 1
      end while
   end for
return num
```

只需遍历数组一遍，时间复杂度为 O(n)

2. 如果在单处理器上，有 n 个运行时间分别为t_1，t_2，$\cdots$，t_n的已知作业，请考虑安排一个调度计划，使得所有作业花费在系统中的时间最少（一个作业花费在系统中的时间是该作业用于等待的时间和用于运行的时间的总和）。为该问题设计一个贪婪算法。讨论所设计的贪婪算法是否总能产生最优解。

答：

首先我们假设只有三个运行时间为 a、b、c 的作业，且 a<b<c

假设让它们顺序工作，那么总时间 T1 为 a+(a+b)+(a+b+c) = 3a+2b+c，

若改变倒序工作，总时间 T2 为 c+(c+b)+(c+b+a) = 3c+2b+a；

T1-T2 = 2(a-c)<0，所以，按短作业优先调度时间最短。

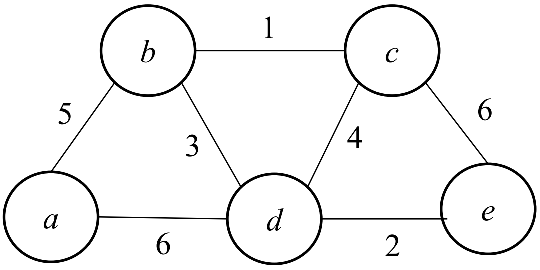
对 n 个作业按执行时间从小到大重新进行排序，则对 n 个作业：运行时间满足：$t_1 \le t_2 \le \ldots\ldots \le t_{(n-1)} \le t_n$。那么有： $T=t_1+(t_1+t_2)+(t_1+t_2+t_3)+\ldots+(t_1+t_2+t_3+\ldots+t_n)$

$=n\times t_1+(n-1)\times t_2+(n-3)\times t_3)+\ldots+t_n$

此时运行总时间 T 为最小值。

由于每个作业的运行时间都是固定的，所以总能产生最优解

3. 应用 Krutkal 算法求下图的最小生成树。



步骤：

首先构造边集数组并排序：

每次从边集数组取出一条边

若该边没有被纳入最小生成树子树中并且没有

如果不与之前构成的子树形成环路，那么就将该边纳入最小生成树中。

若该边与之前构成的子树形成环路，那么就跳过此边。

重复执行上述操作，直到将 n-1 条边纳入最小生成树中。

start	end	weight
b	c	1
d	e	2
b	d	3
c	d	4
a	b	5
a	d	6
c	e	6

首先将 b—c 纳入子树，开始构建，不会形成环路

取出 d—e，不够成环路，纳入子树

取出 b—d，不构成环路，纳入子树

取出 c—d，构成环路，跳过

取出 a—b，不构成环路，纳入子树

取出 a—d，构成环路，跳过

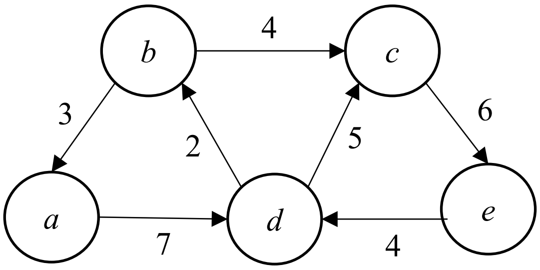
取出 c—e，不构成环路，纳入子树

此时，完成了 n-1 条边的构建，最小生成树构建完成

4. 设计一个求加权连通图的最大生成树算法，其中，最大生成树是包含最大可能权重的树。

将图中的每条边的权重变为其相反数，然后利用 prim 算法或 Kruskal 算法求其最小生成树，之后再把边的权重改回来即可

5. 求解以下单源最短路径问题的实例，以顶点 *a* 作为起点，给出求解过程。



初使时令 S={a},T={其余顶点}，T 中顶点对应的距离值， 若存在<V0,Vi>，为<V0,Vi>弧上的权值（和ＳＰＦＡ初始化方式不同)，若不存在<V0,Vi>，为 Inf。

从 a 出发，只有一条边可走，即 a→d，此时求出 a→d 最短路径为 7，将 d 纳入 S 中。S={a,d}

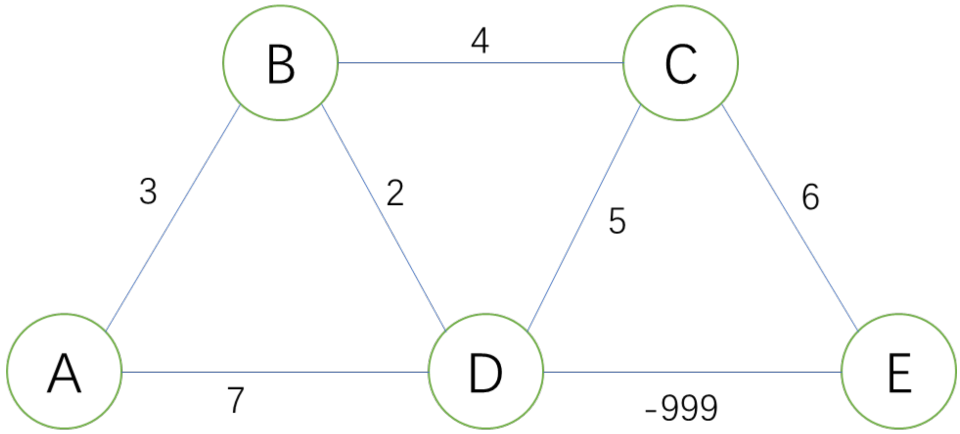
而后再从 d 出发，可走 d→b 和 d→a 两条边，此时更新 a→b 最短路径为 9，a→c 最短路径为 12，将 b 纳入 S 中，S={a,d,b}

从 b 出发，可走 b→a 和 b→c 两条边，由于 a→d→b→c 的权值大于 a→d→c，此时不更新最短路径，将 c 纳入 S 中 S={a,d,b,c}

从 c 出发，可走 c→e 一条边，更新 a→e 最短路径为 18

从 e 出发，可走 e→d 一条路，a→d→c→e→d 权值大于 7，不更新 a→d 最短路径，将 d 纳入 S，S={a,d,b,c,e}。所有顶点已经被纳入 S，算法结束

6. 给出一个反例，说明对于包含负权重的加权连通图，Dijkttra 算法可能会无效。

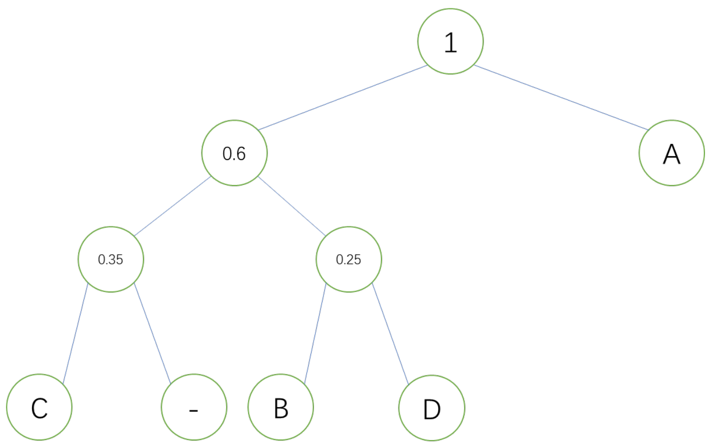


7. （1）对于下面的数据构造哈夫曼编码：

字符	A	B	C	D	-
出现概率	0.4	0.1	0.2	0.15	0.15

（2）用（1）中的编码对文本 ABACABAD 进行编码；

（3）对于编码为 100010111001010 文本用（1）中的编码进行解码。



A:1　　B:010　　C:000　　D:011　　-:001

ABACABAD:1010100010101011

100010111001010:　ACDA-B