

hw5

P3, P5, P11 (a) (b) P11 的第 c 小题不做

P5 (提示：假设采用同步方式计算，给出初始以及之后每一轮的节点 z 的距离表。第 i 轮，节点 z 从邻居收到的距离向量为到目的地最多经过 i 跳的路由，而节点 z 的距离向量为最多经过  $i+1$  跳的路由。)

距离表可采用下述格式描述，前面两行为  $v$  和  $x$  发给节点  $z$  的距离向量，而最后一行  $z$  为节点计算出来的距离向量

目的	u	v	x	y	z
v					
x					
z					

P11 (a) (b)

注意教材 P11 的第 a 小题的中文翻译有误!

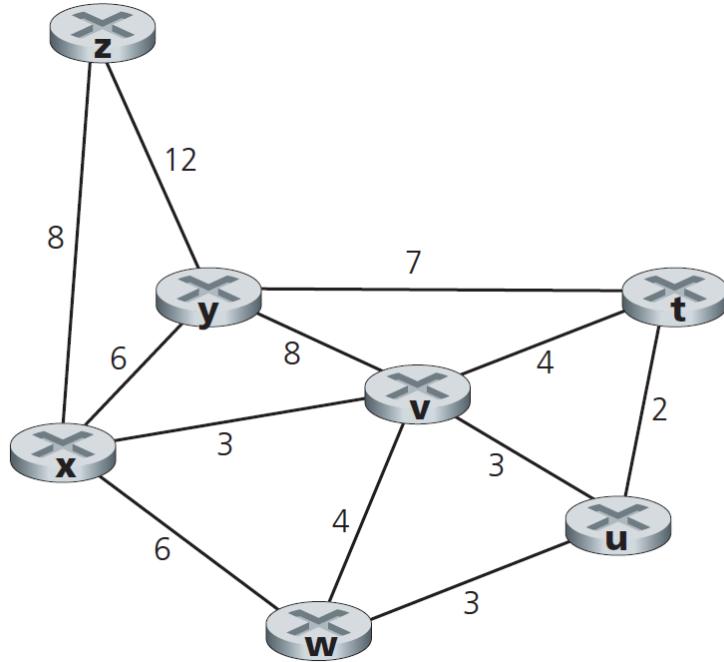
- a. When the distance vector routing is stabilized, router w, y, and z inform their distances to x to each other. What distance values do they tell each other? 教材翻译有误，路由器 w, y 和 z 之间互相告诉对方其到 x 的距离，请问互相告知的到 x 的距离分别为多少？

(可以采用 ppt 中 P22 页的表格描述算法的步骤，仅仅考虑到 x 的路由)

表格中给出节点 y、z、w 的距离表中到 x 的距离向量，比如节点 y 对应三列，第一列为自己的距离向量，给出了距离以及下一跳节点，而后面两列为 z 和 w 发给 y 的距离向量。

Cost那一行给出了节点到邻居的链路花费，或者节点到x的链路花费

P3 Consider the following network. With the indicated link costs, use Dijkstra's shortest-path algorithm to compute the shortest path from x to all network nodes. Show how the algorithm works by computing a table similar to Table 5.1.



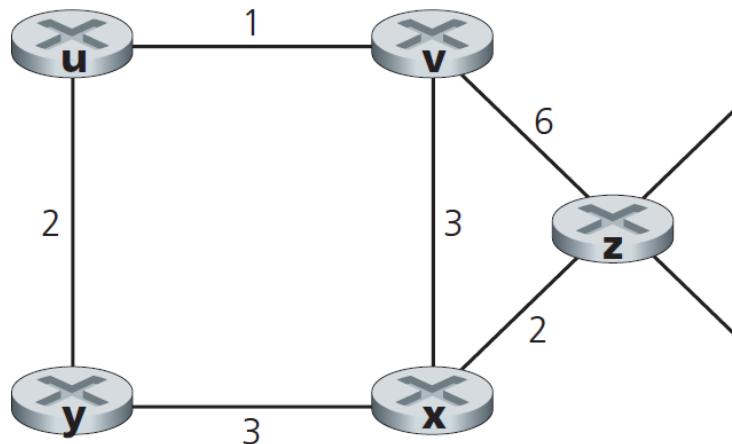
Step	N'	D(t),p(t)	D(u),p(u)	D(v),p(v)	D(w),p(w)	D(y),p(y)	D(z),p(z)
0							
1							
2							
3							
4							
5							
6							

P5 Consider the network shown below, and assume that each node initially knows the costs to each of its neighbors. Consider the distance-vector algorithm and show the distance table entries at node z.

(提示：假设采用同步方式计算，给出初始以及之后每一轮的节点 z 的距离表。第 i 轮，节点 z 从邻居收到的距离向量为到目的地最多经过 i 跳的路由，而节点 z 的距离向量为最多经过 i+1 跳的路由。)

距离表可采用下述格式描述，前面两行为 v 和 x 发给节点 z 的距离向量，而最后一行 z 为节点计算出来的距离向量

目的	u	v	x	y	z
v					
x					
z					



P11 (a) (b) 第 c 小题不做

Consider Figure 5.7. Suppose there is another router w, connected to router y and z. The costs of all links are given as follows:  $c(x,y) = 4$ ,  $c(x,z) = 50$ ,  $c(y,w) = 1$ ,  $c(z,w) = 1$ ,  $c(y,z) = 3$ . Suppose that poisoned reverse is used in the distance-vector routing algorithm.

- a. When the distance vector routing is stabilized, router w, y, and z inform their distances to x to each other. What distance values do they tell each other? 教材翻译有误，路由器 w, y 和 z 之间互相告诉对方其到 x 的距离，请问互相告知的到 x 的距离分别为多少？

b. Now suppose that the link cost between x and y increases to 60. Will there be a count-to-infinity problem even if poisoned reverse is used? Why or why not? If there is a count-to-infinity problem, then how many iterations are needed for the distance-vector routing to reach a stable state again? Justify your answer.

c. How do you modify  $c(y,z)$  such that there is no count-to-infinity problem at all if  $c(y,x)$  changes from 4 to 60?

(可以采用 ppt 中 P22 页的表格描述算法的步骤，仅仅考虑到 x 的路由)

表格中给出节点 y、z、w 的距离表中到 x 的距离向量，比如节点 y 对应三列，第一列为自己的距离向量，给出了距离以及下一跳节点，而后面两列为 z 和 w 发给 y 的距离向量。Cost 那一行给出了节点到邻居的链路花费，或者节点到 x 的链路花费