作业反馈3-8

TC第24.1节练习2、3、**4**

TC第24.2节练习2

TC第24.3节练习2、**4**、7

TC第24.5节练习2、5

TC第24章问题2、3

24.1-4

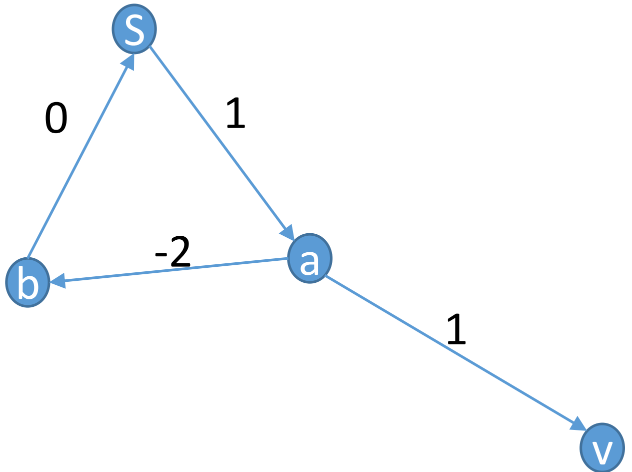
Modify the Bellman-Ford algorithm so that it sets $v.d$ to $-\infty$ for all vertices v for which there is a negative-weight cycle on some path from the source to v.

```
BELLMAN-FORD(G, w, s)
1  INITIALIZE-SINGLE-SOURCE(G, s)
2  for i = 1 to |G.V| - 1
3      for each edge (u, v) ∈ G.E
4          RELAX(u, v, w)
5  for each edge (u, v) ∈ G.E
6      if v.d > u.d + w(u, v)
7          return FALSE
8  return TRUE
```

```
Line 5-7:
  for i = 1 to |G.V| - 1
    for each edge (u, v) in G.E
      if v.d > u.d + w(u, v)
        v.d = -inf
```

$v.d = -\infty$

?



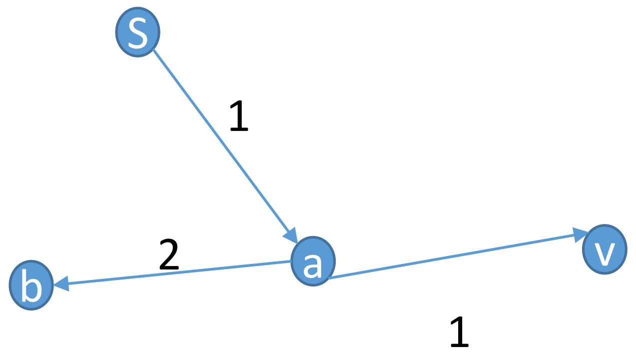
24.3-4

Professor Gaedel has written a program that he claims implements Dijkstra's algorithm. The program produces $v.d$ and $v.\pi$ for each vertex $v \in V$. Give an $O(V+E)$-time algorithm to check the output of the professor's program. It should determine whether the d and π attributes match those of some shortest-paths tree. You may assume that all edge weights are nonnegative.

```
for each vertex v in G
  assert(v.d == v.pi.d + w(v.pi, v))
for each edge (u, v) in G
  assert(v.d <= y.d + w(u, v))
```

是否有问题？

如果s.d=5




```
CHECK-DIJKSTRA(G, w)

        for every vertex u belongs to G

            for every (u, v) belongs to G.E

                if(v.pi != u && u.d + w(u, v) < v.d)

                    return false

                if(v.pi == u && u.d + w(u, v) != v.d)

                    return false

        return true
```

24.3-4

Professor Gaedel has written a program that he claims implements Dijkstra's algorithm. The program produces $v.d$ and $v.\pi$ for each vertex $v \in V$. Give an $O(V + E)$-time algorithm to check the output of the professor's program. It should determine whether the d and π attributes match those of some shortest-paths tree. You may assume that all edge weights are nonnegative.

- 具有唯一的节点s，$s.d = 0, s.\pi = Nil$

```
for each vertex v in G.V-{s}
  assert(v.d == v.pi.d + w(v.pi, v))
for each edge (u, v) in G
  assert(v.d <= y.d + w(u, v))
```

24-2 *Nesting boxes*

A d-dimensional box with dimensions $(x_1, x_2, \ldots, x_d)$ ***nests*** within another box with dimensions $(y_1, y_2, \ldots, y_d)$ if there exists a permutation π on $\{1, 2, \ldots, d\}$ such that $x_{\pi(1)} < y_1, x_{\pi(2)} < y_2, \ldots, x_{\pi(d)} < y_d$.

a. Argue that the nesting relation is transitive.

b. Describe an efficient method to determine whether or not one d-dimensional box nests inside another.

c. Suppose that you are given a set of n d-dimensional boxes $\{B_1, B_2, \ldots, B_n\}$. Give an efficient algorithm to find the longest sequence $\langle B_{i_1}, B_{i_2}, \ldots, B_{i_k} \rangle$ of boxes such that B_{i_j} nests within $B_{i_{j+1}}$ for $j = 1, 2, \ldots, k - 1$. Express the running time of your algorithm in terms of n and d.

a. 假设 $x=(x_1,x_2,\ldots,x_d)$ nests within $y=(y_1,y_2,\ldots,y_d)$, $y=(y_1,y_2,\ldots,y_d)$ nests within $z=(z_1,z_2,\ldots,z_d)$. 则, $\exists \pi_1,\pi_2 \in S_d$(d 阶对称群) $\ni x_{\pi_1(i)} < y_i, y_{\pi_2(i)} < z_i\ \forall i \in \{1,2,\ldots,d\}$. 于是, 有 $x_{\pi_1(\pi_2(i))} < y_{\pi_2(i)} < z_i\ \forall i \in \{1,2,\ldots,d\}$. 因此, $x=(x_1,x_2,\ldots,x_d)$ nests within $z=(z_1,z_2,\ldots,z_d)$.

b. 现将两个数组中的元素从小到大排序, 然后逐个比较大小. 如果 x 的元素逐个比 y 小, 则 x nests in y. 如果使用快速排序, 平均时间复杂度为 $\Theta(d\lg d)$.

c. 可以先将所有的 B_i 两两比较, 确定其 nesting 关系, 此过程的时间复杂度为 $\Theta(n^2 d\lg d)$. 在这个过程中可以构建一个图 $G=(V,E)$, 其中 $V=\{B_1,B_2,\ldots,B_n\}\cup\{S\}$, $E=\{(B_u,B_v)|B_u \text{ nests within } B_v\}\cup\{(S,B_i)|i=1,2,\ldots,n\}$. 每条边的权都是 1. 于是, 问题转化为在 G 中求以 S 为源的关键路径. 求关键路径的时间复杂度为 $\Theta(|V|+|E|)$, 即 $\Theta(n^2)$. 综上, 算法的时间复杂度为 $\Theta(n^2 d\lg d)$.

还有什么办法?

24-3 *Arbitrage*

Arbitrage is the use of discrepancies in currency exchange rates to transform one unit of a currency into more than one unit of the same currency. For example, suppose that 1 U.S. dollar buys 49 Indian rupees, 1 Indian rupee buys 2 Japanese yen, and 1 Japanese yen buys 0.0107 U.S. dollars. Then, by converting currencies, a trader can start with 1 U.S. dollar and buy $49 \times 2 \times 0.0107 = 1.0486$ U.S. dollars, thus turning a profit of 4.86 percent.

Suppose that we are given n currencies $c_1, c_2, \ldots, c_n$ and an $n \times n$ table R of exchange rates, such that one unit of currency c_i buys $R[i, j]$ units of currency c_j.

a. Give an efficient algorithm to determine whether or not there exists a sequence of currencies $\langle c_{i_1}, c_{i_2}, \ldots, c_{i_k} \rangle$ such that

$$R[i_1, i_2] \cdot R[i_2, i_3] \cdots R[i_{k-1}, i_k] \cdot R[i_k, i_1] > 1 \,.$$

Analyze the running time of your algorithm.

b. Give an efficient algorithm to print out such a sequence if one exists. Analyze the running time of your algorithm.

observing that $R[i_1,i_2]\cdot R[i_2,i_3]\ldots R[i_{K-1},i_K]\cdot R[i_K,i_1]>1$. if and only if

$$\frac{1}{R[i_1,i_2]}\cdot\frac{1}{R[i_2,i_3]}\cdots\cdots\frac{1}{R[i_{K-1},i_K]}\cdot\frac{1}{R[i_K,i_1]}<1$$,Taking logs

$$\lg\frac{1}{R[i_1,i_2]}+\lg\frac{1}{R[i_2,i_3]}+\lg\frac{1}{R[i_{K-1},i_K]}+\ldots+\lg\frac{1}{R[i_K,i_1]}<0$$

负权cycle

Therefore if we define the weight of edge (v_i,v_j) as $\omega(v_i,v_j)=\lg\frac{1}{R[i,j]}$

$$=-\lg R[i,j].$$

Bellman Ford

ARBITRAGE(R)

for each pair (i,j)
    $R[i,j] = -\log(R[i,j])$
    $R[j,i] = -\log(1/R[i,j])$
return not BELLMAN-FORD($G, R, 0$)