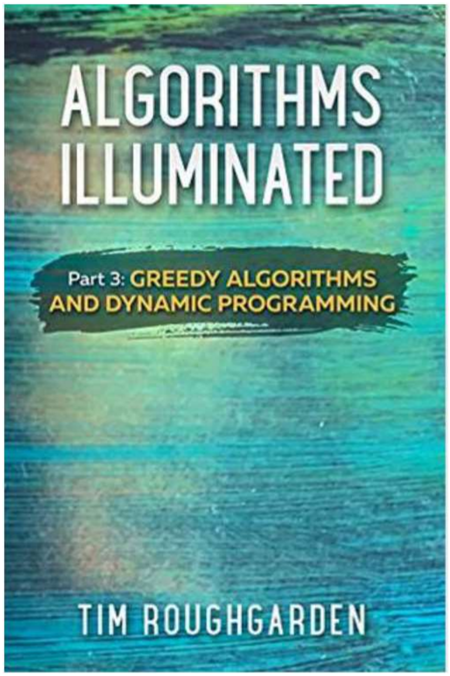
طراحی الگوریتم

(برنامه‌ریزی پویا)

دانشکده مهندسی برق و کامپیوتر، دانشگاه صنعتی اصفهان

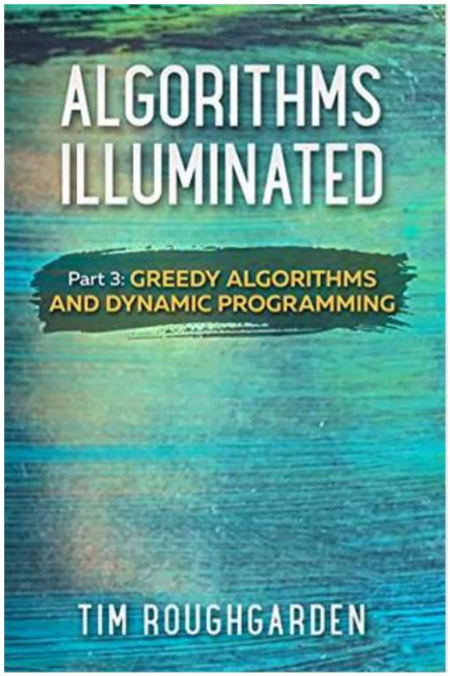
بهار ۱۴۰۰

کوتاهترین فاصله از یک منبع



فصل هجدهم، صفحه ۱۶۷

کوتاهترین فاصله از یک منبع



فصل هفدهم، صفحه ۱۶۷

ورودی: یک گراف جهت‌دار، یک رأس منبع، و برای هر یال یک طول.

هدف: کوتاهترین فاصله برای هر رأس از منبع.

ساختار جواب بهینه

ساختار جواب بهینه

الگوریتم Bellman-Ford

Bellman-Ford

Input: directed graph $G = (V, E)$ in adjacency-list representation, a source vertex $s \in V$, and a real-valued length ℓ_e for each $e \in E$.
Output: $dist(s, v)$ for every vertex $v \in V$, or a declaration that G contains a negative cycle.

```
// subproblems (i indexed from 0, v indexes V)
A := (n + 1) × n two-dimensional array
// base cases (i = 0)
A[0][s] := 0
for each v ≠ s do
    A[0][v] := +∞
// systematically solve all subproblems
for i = 1 to n do                  // subproblem size

    for v ∈ V do
        // use recurrence from Corollary 18.2
        A[i][v] :=
          min{A[i − 1][v], min_{(w,v)∈E} {A[i − 1][w] + ℓ_wv}}
```

$$\min\{\underbrace{A[i-1][v]}_{\text{Case 1}}, \underbrace{\min_{(w,v)\in E}\{A[i-1][w] + \ell_{wv}\}}_{\text{Case 2}}\}$$

نتیجه :

$L_{i,v} =$ کوتاهترین فاصله s به v با حداکثر i یال

$+\infty =$ تا v اگر چنین مسیری وجود نداشته باشد.

$$\boxed{L_{i,v} = \min \begin{cases} L_{i-1,v} & \text{حالت (1)} \\ \min\limits_{(w,v)\in E} \{L_{i-1,w} + \ell_{wv}\} & \text{حالت (2)} \end{cases}}$$

$$L_{0,v} = \begin{cases} 0 & v = s \\ +\infty & v \neq s \end{cases}$$

الگوریتم Bellman-Ford

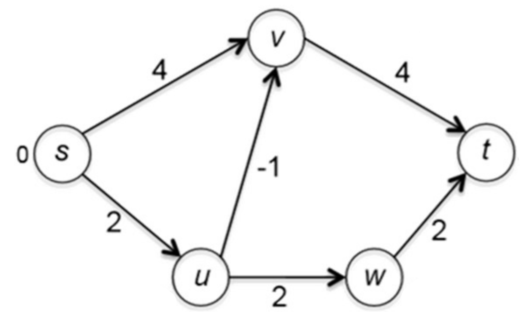
فعلا فرض کنید دور منفی نداریم.

Bellman-Ford

Input: directed graph $G = (V, E)$ in adjacency-list representation, a source vertex $s \in V$, and a real-valued length ℓ_e for each $e \in E$.
Output: $dist(s, v)$ for every vertex $v \in V$, or a declaration that G contains a negative cycle.

// subproblems (i indexed from 0, v indexes V)
$A := (n+1) \times n$ two-dimensional array
// base cases ($i = 0$)
$A[0][s] := 0$
for each $v \neq s$ **do**
  $A[0][v] := +\infty$
// systematically solve all subproblems
for $i = 1$ to n **do** // subproblem size

  for $v \in V$ **do**
    // use recurrence from Corollary 18.2
    $A[i][v] :=$
     $\min\{\underbrace{A[i-1][v]}_{\text{Case 1}}, \underbrace{\min_{(w,v)\in E}\{A[i-1][w] + \ell_{wv}\}}_{\text{Case 2}}\}$



الگوریتم Bellman-Ford

Bellman-Ford

Input: directed graph $G = (V, E)$ in adjacency-list representation, a source vertex $s \in V$, and a real-valued length ℓ_e for each $e \in E$.
Output: $dist(s, v)$ for every vertex $v \in V$, or a declaration that G contains a negative cycle.

`// subproblems (`i` indexed from 0, `v` indexes `V`)`
$A := (n+1) \times n$ two-dimensional array
`// base cases (`$i = 0$`)`
$A[0][s] := 0$
for each $v \neq s$ **do**
    $A[0][v] := +\infty$
`// systematically solve all subproblems`
for $i = 1$ to n **do** `// subproblem size`
    stable := TRUE `// for early stopping`
    for $v \in V$ **do**
        `// use recurrence from Corollary 18.2`
        $A[i][v] :=$
$$\min\{\underbrace{A[i-1][v]}_{\text{Case 1}}, \underbrace{\min_{(w,v)\in E}\{A[i-1][w] + \ell_{wv}\}}_{\text{Case 2}}\}$$
        if $A[i][v] \neq A[i-1][v]$ **then**
            stable := FALSE
    if stable = TRUE **then**
        return $\{A[i-1][v]\}_{v \in V}$
`// failed to stabilize in `n` iterations`
return "negative cycle"