Análise e Síntese de Algoritmos

Fluxos Máximos [CLRS, Cap. 26]

2011/2012

Contexto

- Revisão [CLRS, Cap.1-13]
  - Fundamentos; notação; exemplos
- Algoritmos em Grafos [CLRS, Cap.21-26]
  - Algoritmos elementares
  - Árvores abrangentes
  - Caminhos mais curtos
  - Fluxos máximos
- Programação Linear [CLRS, Cap.29]
  - Algoritmos e modelação de problemas com restrições lineares
- Técnicas de Síntese de Algoritmos [CLRS, Cap.15-16]
  - Programação dinâmica
  - Algoritmos greedy
- Tópicos Adicionais [CLRS, Cap.32-35]
  - Emparelhamento de Cadeias de Caracteres
  - Complexidade Computacional
  - Algoritmos de Aproximação

Resumo

Intuição

Pré-Fluxos - Intuição

- Operações mais localizadas do que Ford-Fulkerson
  - Não é baseada na identificação de caminhos de aumento

Intuição

Pré-Fluxos - Intuição

- Operações mais localizadas do que Ford-Fulkerson
  - Não é baseada na identificação de caminhos de aumento
- Propriedade da conservação de fluxo não é mantida durante execução do algoritmo

Intuição

Pré-Fluxos - Intuição

- Operações mais localizadas do que Ford-Fulkerson
  - Não é baseada na identificação de caminhos de aumento
- Propriedade da conservação de fluxo não é mantida durante execução do algoritmo
- Cada vértice u contém reservatório de fluxo
  - Representa excesso de fluxo $e(u)$
  - Começar por enviar todo o fluxo possível de s para vértices adjacentes

Intuição

Pré-Fluxos - Intuição

- Operações mais localizadas do que Ford-Fulkerson
  - Não é baseada na identificação de caminhos de aumento
- Propriedade da conservação de fluxo não é mantida durante execução do algoritmo
- Cada vértice u contém reservatório de fluxo
  - Representa excesso de fluxo $e(u)$
  - Começar por enviar todo o fluxo possível de s para vértices adjacentes
- Noção de altura de cada vértice, que evolui com aplicação do algoritmo
  - Envio de fluxo só de vértices mais altos para vértices mais baixos
  - Fazer subir altura de vértices em caso de necessidade de envio de fluxo

Definições

Pré-Fluxos - Definições

- Pré-Fluxo: $f : V \times V \rightarrow R$
  - Verifica restrições de capacidade, simetria e $f(V, u) \geq 0$ para vértices $u \in V - \{s\}$
  - Não verifica necessariamente conservação de fluxo

Definições

Pré-Fluxos - Definições

- Pré-Fluxo: $f : V \times V \to R$
  - Verifica restrições de capacidade, simetria e $f(V, u) \geq 0$ para vértices $u \in V - \{s\}$
  - Não verifica necessariamente conservação de fluxo
- Excesso de fluxo: $e(u) = f(V, u)$
  - $u \in V - \{s, t\}$ transborda se $e(u) > 0$

Definições

Pré-Fluxos - Definições

- Pré-Fluxo: $f : V \times V \to R$
  - Verifica restrições de capacidade, simetria e $f(V, u) \geq 0$ para vértices $u \in V - \{s\}$
  - Não verifica necessariamente conservação de fluxo
- Excesso de fluxo: $e(u) = f(V, u)$
  - $u \in V - \{s, t\}$ transborda se $e(u) > 0$
- Uma função $h : V \to N$ é uma função de alturas se $h(s) = |V|$, $h(t) = 0$, e $h(u) \leq h(v) + 1$ para todo o arco residual $(u, v) \in E_f$
  - Função de alturas permite estabelecer condições para ser possível enviar fluxo de u para v

Operações Básicas

Envio de fluxo de u para v

Push(u, v)

1 $d_f(u, v) = min(e[u], c_f[u, v])$
2 $f[u, v] = f[u, v] + d_f(u, v)$
3 $f[v, u] = -f[u, v]$
4 $e[u] = e[u] - d_f(u, v)$
5 $e[v] = e[v] + d_f(u, v)$

Aplica-se quando u transborda, $c_f[u, v] > 0$, e $h[u] = h[v] + 1$

Subir a altura de u

Relabel(u)

1 $h[u] = 1 + \min\{h[v] : (u, v) \in E_f\}$

Aplica-se quando u transborda, e $(u, v) \in E_f$ implica $h[u] \leq h[v]$

Operações Básicas

Pré-Fluxos - Operações Básicas

Operação de envio de fluxo de u para v, Push(u, v):

- Saturating push: arco (u, v) fica saturado após aplicação da operação Push (i.e., $f(u, v) = c(u, v)$ e $c_f(u, v) = 0$)
- Caso contrário: Nonsaturating push
- OBS: Após um nonsaturating Push(u, v), u deixa de transbordar (i.e., $e(u) = 0$)

Operações Básicas

Inicialização

Initialize-PreFlow(G, s)

for each $v \in V[G]$
  do $h[u] = 0$
    $e[u] = 0$
for each $(u, v) \in E[G]$
  do $f[u, v] = 0$
    $f[v, u] = 0$
$h[s] = |V[G]|$
for each $u \in Adj[s]$
  do $f[s, u] = c(s, u)$
    $f[u, s] = -c(s, u)$
    $e[u] = c(s, u)$

Algoritmo Genérico

Algoritmo Genérico

Generic-Push-Relabel(G, s)

1 Initialize-Preflow(G, s)
2 **while** existe operação de Push ou Relabel aplicável
3 **do** seleccionar e executar operação de Push ou Relabel
4 **return** f

Correcção do Método Genérico de Pré-Fluxos

Correcção do Método (1)

Seja $G = (V, E)$ uma rede de fluxo com fonte s e destino t, f um pré-fluxo, e h uma função de alturas para f.
Se vértice u transborda, então u pode ser sujeito a uma operação de Relabel ou de Push

- h é função de alturas, pelo que $h(u) \leq h(v) + 1$
- Se operação de Push não aplicável a u, então para qualquer arco residual $(u, v) \in E_f$, $h(u) < h(v) + 1$, pelo que $h(u) \leq h(v)$
- Assim, operação de Relabel pode ser aplicada a u

Correcção do Método Genérico de Pré-Fluxos

Correcção do Método (2)

A altura dos vértices nunca decresce.
Se operação de Relabel é aplicada, $h[u]$ aumenta de pelo menos 1 unidade

- Valor de $h[u]$ apenas alterado em Relabel
- Aplicar Relabel(u) se:
  - $\forall(u, v) \in E_f : h[u] \leq h[v]$
- $h[u] < 1 + \min\{h[v] : (u, v) \in E_f\}$, antes de Relabel
- Pelo que valor de $h[u]$ aumenta (de pelo menos 1 unidade) após Relabel

Correcção do Método Genérico de Pré-Fluxos

Correcção do Método (3)

Durante a execução do algoritmo genérico o valor de h é mantido como função de alturas

- Inicialmente h é uma função de alturas

Correcção do Método Genérico de Pré-Fluxos

Correcção do Método (3)

Durante a execução do algoritmo genérico o valor de h é mantido como função de alturas

- Inicialmente h é uma função de alturas
- Relabel(u) mantém h como função de alturas
  - Para os arcos (u, v) em E_f
    - $h[u] \leq h[v] + 1$ após Relabel, pela definição de Relabel
  - Para os arcos (w, u) em E_f
    - $h[w] \leq h[u] + 1$ antes de Relabel implica $h[w] < h[u] + 1$ após Relabel de u

Correcção do Método Genérico de Pré-Fluxos

Correcção do Método (3)

Durante a execução do algoritmo genérico o valor de h é mantido como função de alturas

- Inicialmente h é uma função de alturas
- Relabel(u) mantém h como função de alturas
  - Para os arcos (u, v) em E_f
    - $h[u] \leq h[v] + 1$ após Relabel, pela definição de Relabel
  - Para os arcos (w, u) em E_f
    - $h[w] \leq h[u] + 1$ antes de Relabel implica $h[w] < h[u] + 1$ após Relabel de u
- Push(u, v) mantém h como função de alturas
  - Arco (v, u) fica incluído em E_f
    - $h[v] = h[u] - 1 < h[u] + 1$
  - Se arco (u, v) é removido de E_f
    - deixa de existir restrição em h devido a (u, v)

Correcção do Método Genérico de Pré-Fluxos

Correcção do Método (Resumo)

- Se vértice u transborda, então u pode ser sujeito a uma operação de Relabel ou de Push
- A altura dos vértices nunca decresce. Se operação de Relabel é aplicada, $h[u]$ aumenta de pelo menos 1 unidade
- Durante a execução do algoritmo genérico o valor de h é mantido como função de alturas

Correcção do Método Genérico de Pré-Fluxos

Correcção do Método (4)

Na rede residual G_f nunca existe caminho de s para t

- Prova por contradição
- Admitir caminho $p = < v_0, v_1, \ldots, v_k >$ de s para t em G_f, com $v_0 = s$ e $v_k = t$
  - Podemos admitir que p é caminho simples, $k < |V|$
- $i = 0, 1, \ldots, k-1$
  - $(v_i, v_{i+1}) \in E_f$, e $h[v_i] \leq h[v_{i+1}] + 1$ (função de alturas)
- Pelo que, $h[s] \leq h[t] + k$
- Como $h[t] = 0$, então $h[s] \leq k < |V|$
- Mas $h[s] = |V|$; uma contradição !

Correcção do Método Genérico de Pré-Fluxos

Correcção do Método (5)

Se algoritmo genérico termina, o pré-fluxo calculado é fluxo máximo para G

- Inicialmente temos um pré-fluxo
  - Devido a Initialize-Preflow

Correcção do Método Genérico de Pré-Fluxos

Correcção do Método (5)

Se algoritmo genérico termina, o pré-fluxo calculado é fluxo máximo para G

- Inicialmente temos um pré-fluxo
  - Devido a Initialize-Preflow
- Algoritmo mantém a existência de pré-fluxo invariante
  - Push e Relabel não alteram invariante

Correcção do Método Genérico de Pré-Fluxos

Correcção do Método (5)

Se algoritmo genérico termina, o pré-fluxo calculado é fluxo máximo para G

- Inicialmente temos um pré-fluxo
  - Devido a Initialize-Preflow
- Algoritmo mantém a existência de pré-fluxo invariante
  - Push e Relabel não alteram invariante
- Se algoritmo termina, $e[u] = 0$ para qualquer vértice u
  - Caso contrário poderia aplicar-se Push ou Relabel !

Correcção do Método Genérico de Pré-Fluxos

Correcção do Método (5)

Se algoritmo genérico termina, o pré-fluxo calculado é fluxo máximo para G

- Inicialmente temos um pré-fluxo
  - Devido a Initialize-Preflow
- Algoritmo mantém a existência de pré-fluxo invariante
  - Push e Relabel não alteram invariante
- Se algoritmo termina, $e[u] = 0$ para qualquer vértice u
  - Caso contrário poderia aplicar-se Push ou Relabel !
- Nesta situação, pré-fluxo é um fluxo
  - Porque não existem vértices a transbordar !

Correcção do Método Genérico de Pré-Fluxos

Correcção do Método (5)

Se algoritmo genérico termina, o pré-fluxo calculado é fluxo máximo para G

- Inicialmente temos um pré-fluxo
  - Devido a Initialize-Preflow
- Algoritmo mantém a existência de pré-fluxo invariante
  - Push e Relabel não alteram invariante
- Se algoritmo termina, $e[u] = 0$ para qualquer vértice u
  - Caso contrário poderia aplicar-se Push ou Relabel !
- Nesta situação, pré-fluxo é um fluxo
  - Porque não existem vértices a transbordar !
- E não existe caminho de s para t na rede residual
  - Porque h é função de alturas !

Correcção do Método Genérico de Pré-Fluxos

Correcção do Método (5)

Se algoritmo genérico termina, o pré-fluxo calculado é fluxo máximo para G

- Inicialmente temos um pré-fluxo
  - Devido a Initialize-Preflow
- Algoritmo mantém a existência de pré-fluxo invariante
  - Push e Relabel não alteram invariante
- Se algoritmo termina, $e[u] = 0$ para qualquer vértice u
  - Caso contrário poderia aplicar-se Push ou Relabel !
- Nesta situação, pré-fluxo é um fluxo
  - Porque não existem vértices a transbordar !
- E não existe caminho de s para t na rede residual
  - Porque h é função de alturas !
- Pelo teorema do fluxo máximo corte mínimo, f é o fluxo máximo !!

Análise do Método Genérico de Pré-Fluxos

Altura máxima dos vértices

- Para cada vértice u que transborda existe um caminho simples de u para s em G_f
  - OBS: Fluxo enviado tem de poder ser cancelado
- $h[u] \leq 2|V| - 1$ para qualquer $u \in V$
  - $h[s]$ e $h[t]$ são constantes
  - Relabel a u apenas aplicado quando vértice u transborda
  - Existe caminho simples p de u para s
    - $p =< v_0, v_1, \ldots, v_k >$, $v_0 = u$, $v_k = s$, $k \leq |V| - 1$
  - $h[v_i] \leq h[v_{i+1}] + 1, i = 0, 1, \ldots, k - 1$
  - $h[u] = h[v_0] \leq h[v_k] + k \leq h[s] + (|V| - 1) = 2|V| - 1$

Análise do Método Genérico de Pré-Fluxos

Número de operações de Relabel

O número de operações de Relabel é não superior a $2|V|-1$ para cada vértice e a $(2|V|-1)(|V|-2) < 2|V|^2$ no total

- Relabel apenas pode ser aplicado a vértices em $V - \{s,t\}$, i.e. $|V|-2$ vértices
- Relabel faz subir valor de $h[u]$ em pelo menos 1 unidade
- Para $u \in V - \{s,t\}$, valores possíveis para $h[u]$ entre 0 e $2|V|-1$
- Relabel aplicado a u não mais do que $2|V|-1$ vezes
- Número total de operações de Relabel não superior a: $(2|V|-1)(|V|-2) < 2|V|^2$

Análise do Método Genérico de Pré-Fluxos

Número de operações de Saturating Push

O número de saturating pushes é inferior a $2|V||E|$

- Analisar saturating pushes de u para v e de v para u
  - Após Push(u, v), Push(v, u) requer aumento em $h[v]$ de pelo menos 2 unidades
  - Como $0 \leq h[v] \leq 2|V| - 1$, o número máximo de vezes que a altura de v pode aumentar é $|V|$
  - Esses $|V|$ aumentos de $h[v]$ podem implicar o mesmo número de aumentos de $h[u]$, portanto para o par de vértices (u, v) o número total de saturating pushes é inferior a $2|V|$
- Se considerarmos todos os pares de vértices, temos então que o número de saturating pushes é limitado a $2|V||E|$

Análise do Método Genérico de Pré-Fluxos

Número de operações de Non-Saturating Push

O número de nonsaturating pushes é limitado a $4|V|^2(|V|+|E|)$

- Seja $X \subseteq V$ o conjunto de vértices que transborda
- Seja $\Phi = \sum_{v \in X} h[v]$

Análise do Método Genérico de Pré-Fluxos

Número de operações de Non-Saturating Push

O número de nonsaturating pushes é limitado a $4|V|^2(|V| + |E|)$

- Seja $X \subseteq V$ o conjunto de vértices que transborda
- Seja $\Phi = \sum_{v \in X} h[v]$
- Cada operação de Relabel(u) aumenta Φ em menos de $2|V|$
  - Limitação da máxima altura possível para um vértice

Análise do Método Genérico de Pré-Fluxos

Número de operações de Non-Saturating Push

O número de nonsaturating pushes é limitado a $4|V|^2(|V|+|E|)$

- Seja $X \subseteq V$ o conjunto de vértices que transborda
- Seja $\Phi = \sum_{v \in X} h[v]$
- Cada operação de Relabel(u) aumenta Φ em menos de $2|V|$
  - Limitação da máxima altura possível para um vértice
- Cada Saturating Push aumenta Φ em menos de $2|V|$
  - Apenas um novo vértice pode ficar a transbordar e alturas não variam

Análise do Método Genérico de Pré-Fluxos

Número de operações de Non-Saturating Push

O número de nonsaturating pushes é limitado a $4|V|^2(|V|+|E|)$

- Seja $X \subseteq V$ o conjunto de vértices que transborda
- Seja $\Phi = \sum_{v \in X} h[v]$
- Cada operação de Relabel(u) aumenta Φ em menos de $2|V|$
    - Limitação da máxima altura possível para um vértice
- Cada Saturating Push aumenta Φ em menos de $2|V|$
    - Apenas um novo vértice pode ficar a transbordar e alturas não variam
- Non-Saturating Push (u, v) decrementa Φ em pelo menos 1
    - u deixa de transbordar; v pode passar a transbordar e $h[v] - h[u] = -1$

Análise do Método Genérico de Pré-Fluxos

Número de operações de Non-Saturating Push

O número de nonsaturating pushes é limitado a $4|V|^2(|V|+|E|)$

- Seja $X \subseteq V$ o conjunto de vértices que transborda
- Seja $\Phi = \sum_{v \in X} h[v]$
- Cada operação de Relabel(u) aumenta Φ em menos de $2|V|$
  - Limitação da máxima altura possível para um vértice
- Cada Saturating Push aumenta Φ em menos de $2|V|$
  - Apenas um novo vértice pode ficar a transbordar e alturas não variam
- Non-Saturating Push (u, v) decrementa Φ em pelo menos 1
  - u deixa de transbordar; v pode passar a transbordar e $h[v] - h[u] = -1$
- O total de aumento de Φ é limitado a $2|V|(2|V|^2) + 2|V|(2|V||E|) = 4|V|^2(|V| + |E|)$
- Logo, como $\Phi \geq 0$ e no final do algoritmo temos $\Phi = 0$, então $4|V|^2(|V| + |E|)$ é limite superior de nonsaturating pushes.

Análise do Método Genérico de Pré-Fluxos

Complexidade do Método Genérico

- Complexidade é definida pelo número de operações básicas
  - Relabel: $O(V^2)$
  - Saturatiing Pushes: $O(VE)$
  - Non-Saturating Pushes: $O(V^2E)$
- Logo, complexidade do algoritmo genérico é $O(V^2E)$
  - $O(V)$ para operação de Relabel
  - $O(1)$ para operação de Push

Algoritmo Relabel-To-Front

Algoritmo Relabel-To-Front

- Complexidade: $O(V^3)$
- Noção de descarga de um vértice u:
  - Enviar todo o fluxo em excesso para os vértices vizinhos de u
- Lista de vizinhos de u: $N[u]$
  - v em lista $N[u]$ se: $(u, v) \in E$ ou $(v, u) \in E$, i.e. vértices para os quais um arco residual (u, v) pode existir
  - Primeiro vizinho: $head[N[u]]$
  - Próximo vizinho de u (a seguir a v): $next\_neighbor[v]$

Algoritmo Relabel-To-Front

Descarga de um vértice

Discharge(u)

```
while (e[u] > 0)
      do v = current[u]
          if v = NIL
             then Relabel(u)
                     current[u] = head[N[u]]
             else if c_f(u, v) > 0 and h[u] = h[v] + 1
                     then Push(u, v)
                     else current[u] = next_neighbor[v]
```

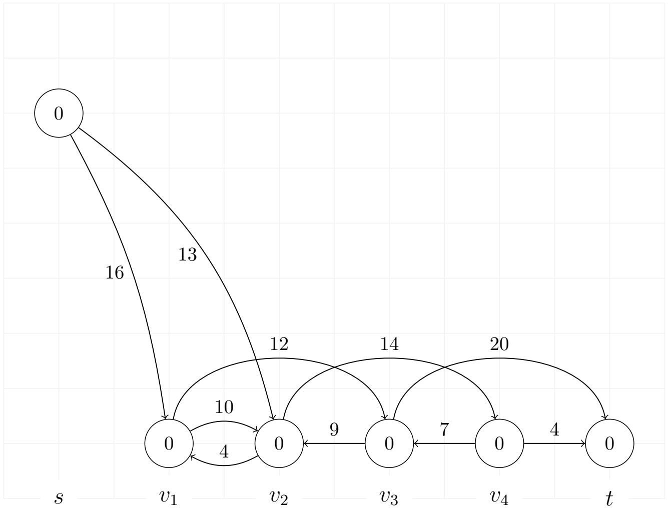
Algoritmo Relabel-To-Front

Relabel-To-Front

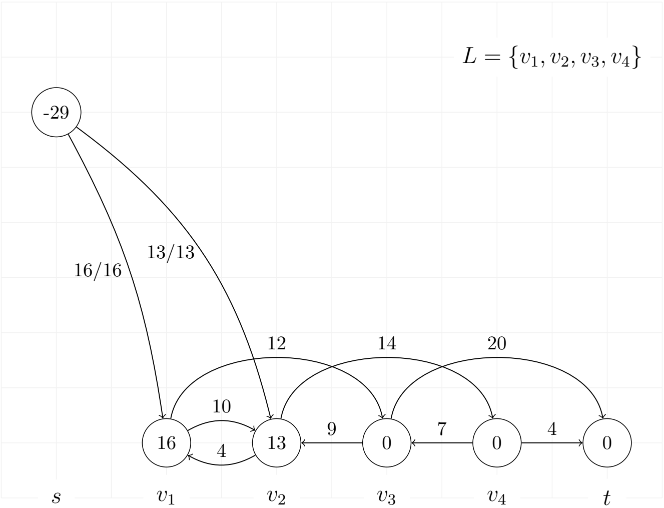
Relabel-To-Front(G, s, t)

Initialize-Preflow(G, s)
$L = V - \{s, t\}$ por qualquer ordem
for each $u \in V - \{s, t\}$
    do $current[u] = head[N[u]]$
$u = head[L]$
while $u \neq$ NIL
    do $oldh = h[u]$
        Discharge(u)
        if $h[u] > oldh$
            then colocar u na frente da lista L
        $u = next[u]$
return f

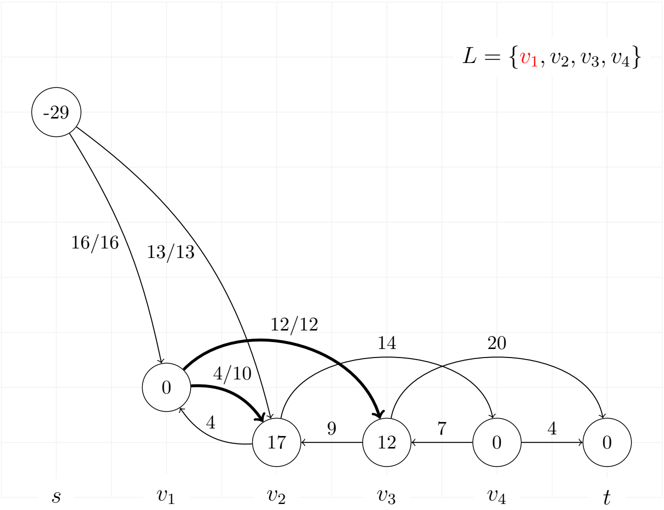
Algoritmo Relabel-To-Front



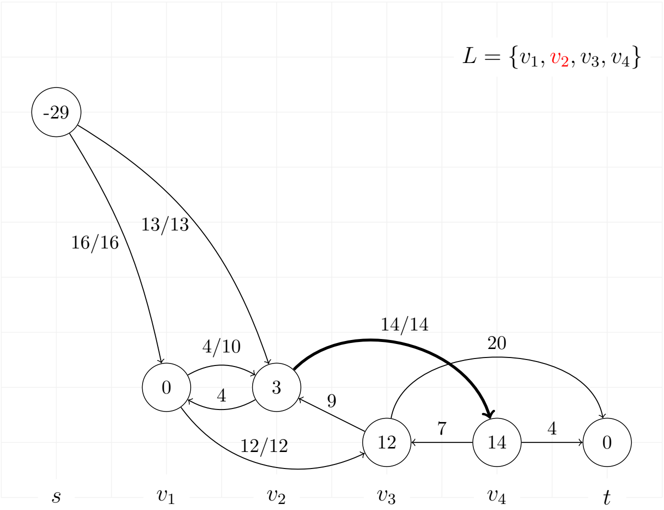
Algoritmo Relabel-To-Front



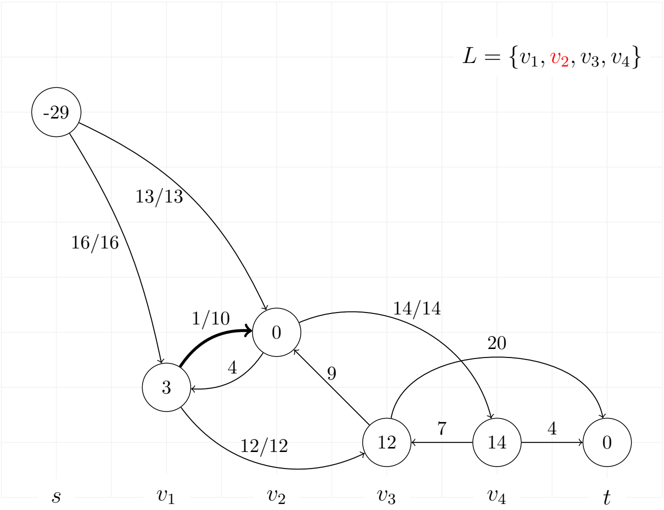
Algoritmo Relabel-To-Front



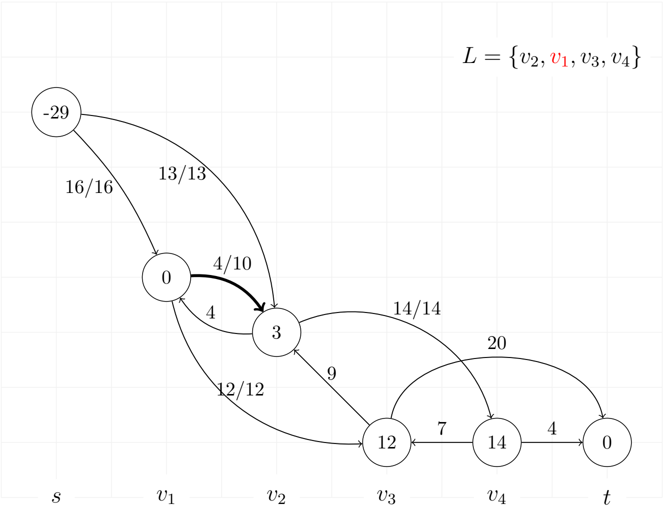
Algoritmo Relabel-To-Front



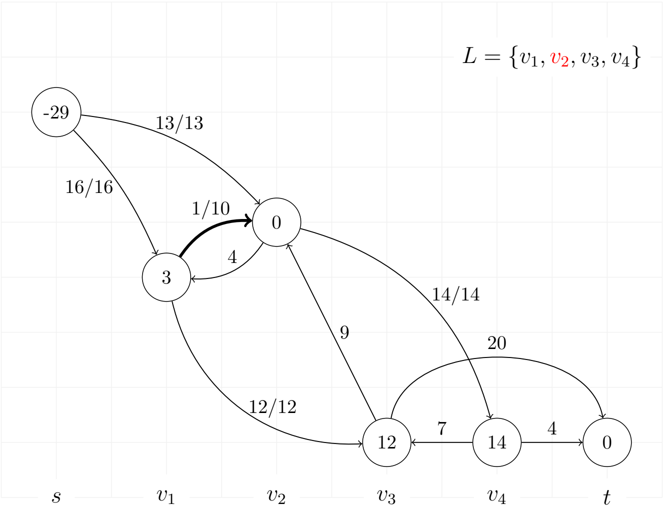
Algoritmo Relabel-To-Front



Algoritmo Relabel-To-Front

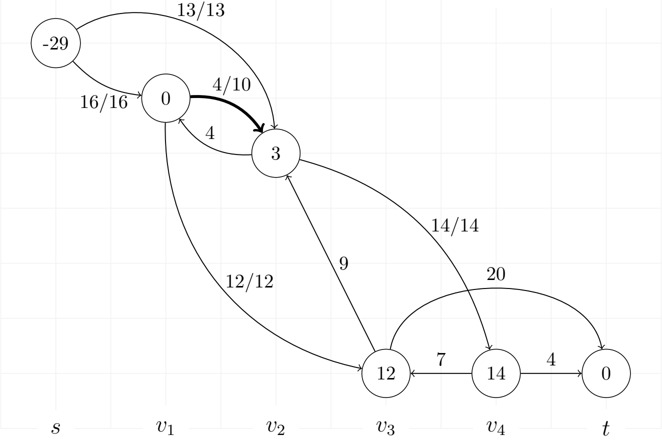


Algoritmo Relabel-To-Front

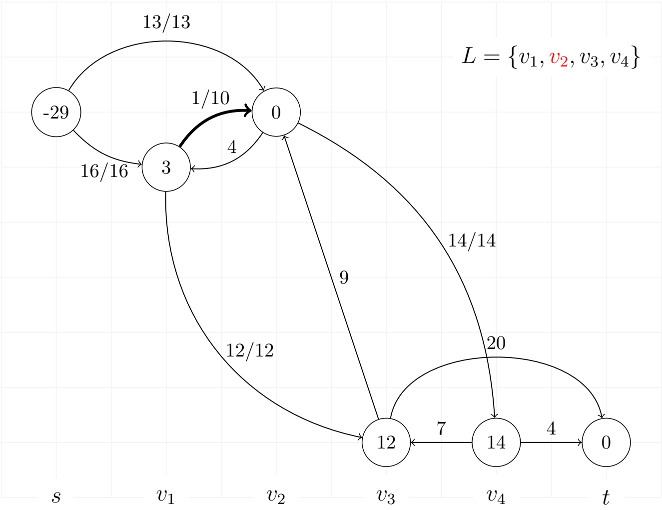


Algoritmo Relabel-To-Front

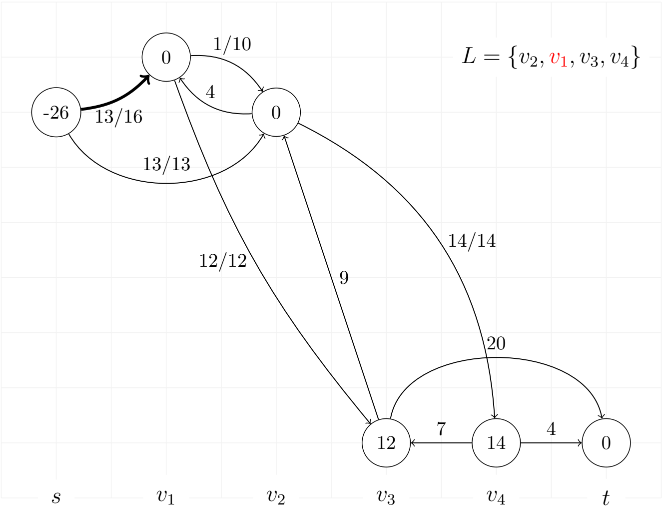
$L = \{v_2, v_1, v_3, v_4\}$



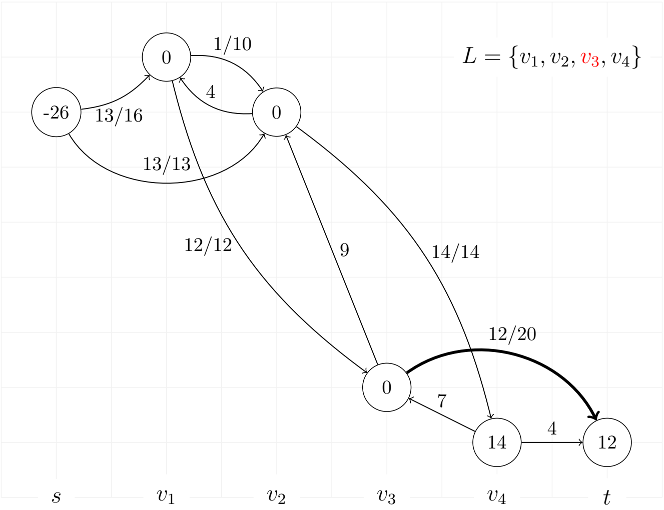
Algoritmo Relabel-To-Front



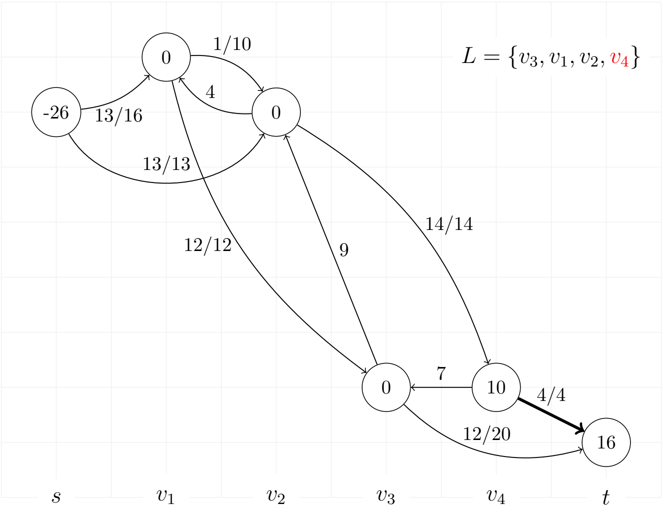
Algoritmo Relabel-To-Front



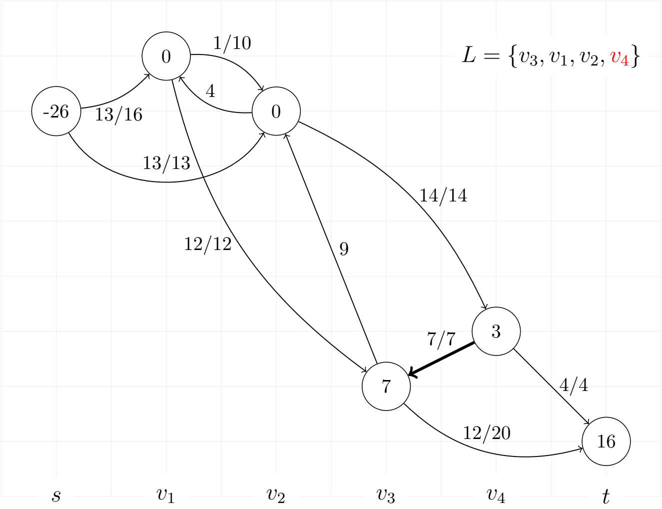
Algoritmo Relabel-To-Front



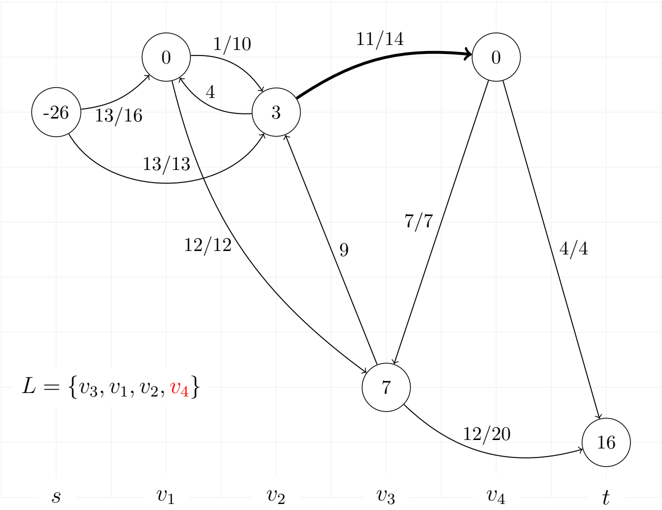
Algoritmo Relabel-To-Front



Algoritmo Relabel-To-Front

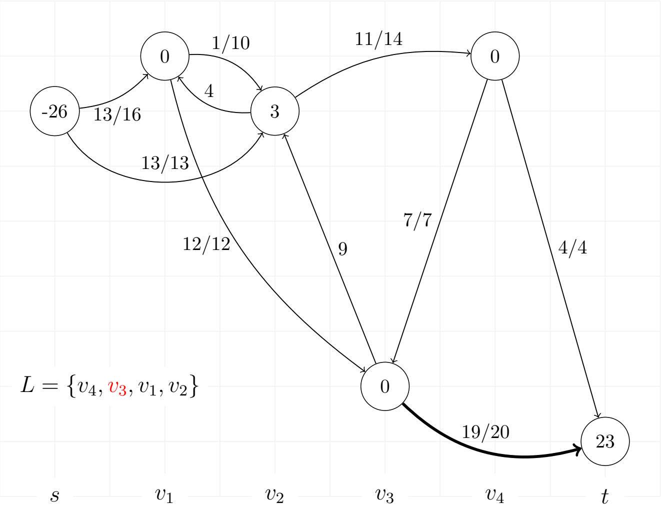


Algoritmo Relabel-To-Front



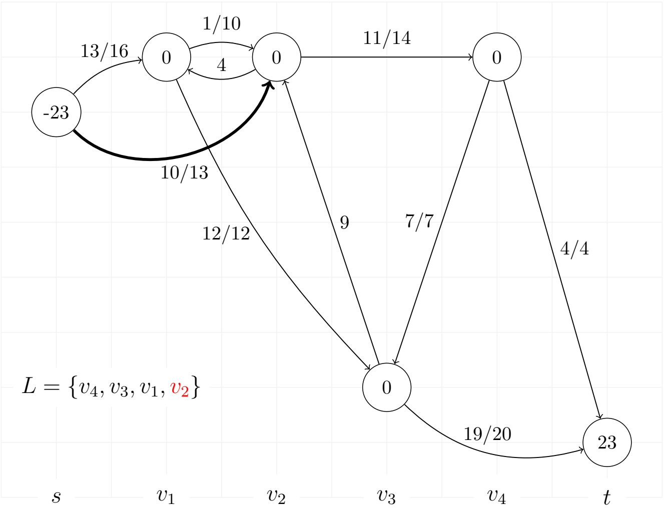
$L = \{v_3, v_1, v_2, v_4\}$

Algoritmo Relabel-To-Front



$L = \{v_4, v_3, v_1, v_2\}$

Algoritmo Relabel-To-Front



Análise Algoritmo Relabel-To-Front

Complexidade

Complexidade do ciclo principal: $O(V^3)$

- Não contabilizando o tempo de Discharge
- Fase: tempo entre operações de Relabel
- Número de fases = número de operações de Relabel = $O(V^2)$
  - $O(V^2)$ para qualquer algoritmo de Pré-Fluxo
- Cada fase consiste de $O(V)$ execuções de Discharge
  - Total de execuções de Discharge é $O(V^3)$
- Complexidade (sem contabilizar Discharge) é $O(V^3)$

Análise Algoritmo Relabel-To-Front

Complexidade

Complexidade acumulada das operações de Discharge:

- Operações Relabel: $O(VE)$ para $O(V^2)$ operações de Relabel

Análise Algoritmo Relabel-To-Front

Complexidade

Complexidade acumulada das operações de Discharge:

- Operações Relabel: $O(VE)$ para $O(V^2)$ operações de Relabel
- Actualizações de $current[u]$:
  - Executadas $O(degree(u))$ vezes após Relabel de u
  - Executadas $O(V\ degree(u))$ no total para cada vértice u (cada vértice pode ser sujeito a O(V) operações de relabel)
  - Total: $O(VE)$

Análise Algoritmo Relabel-To-Front

Complexidade

Complexidade acumulada das operações de Discharge:

- Operações Relabel: $O(VE)$ para $O(V^2)$ operações de Relabel
- Actualizações de $current[u]$:
  - Executadas $O(degree(u))$ vezes após Relabel de u
  - Executadas $O(V\ degree(u))$ no total para cada vértice u (cada vértice pode ser sujeito a O(V) operações de relabel)
  - Total: $O(VE)$
- Saturating pushes: $O(VE)$

Análise Algoritmo Relabel-To-Front

Complexidade

Complexidade acumulada das operações de Discharge:

- Operações Relabel: $O(VE)$ para $O(V^2)$ operações de Relabel
- Actualizações de $current[u]$:
  - Executadas $O(degree(u))$ vezes após Relabel de u
  - Executadas $O(V\ degree(u))$ no total para cada vértice u (cada vértice pode ser sujeito a O(V) operações de relabel)
  - Total: $O(VE)$
- Saturating pushes: $O(VE)$
- Non-saturating pushes:
  - Limitado pelo número de operações Discharge, porque retorna após non-saturating push, i.e. $O(V^3)$

Análise Algoritmo Relabel-To-Front

Complexidade

- Complexidade (total) das operações de Discharge: $O(V^3)$
- Complexidade do algoritmo sem operações de Discharge: $O(V^3)$

- Complexidade do algoritmo Relabel-To-Front: $O(V^3)$