Fluxo máximo

Prof. Dr. Wesin Ribeiro

Neste capítulo

- Introdução
- Redes de fluxo máximo
- O método Ford-Fulkerson
- O Algoritmo push-relabel
- O Algoritmo relabel-to-front
- revisão

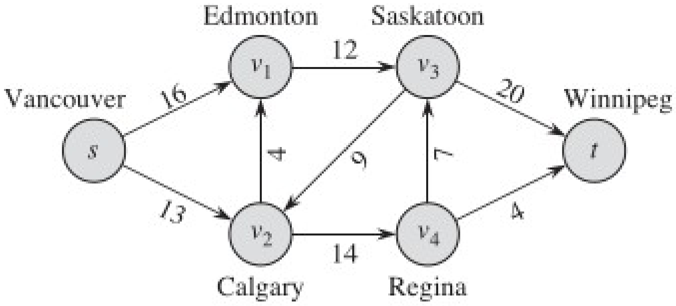
Introdução

Da mesma maneira que podemos modelar um mapa rodoviário como um grafo dirigido para encontrar o caminho mínimo de um ponto a outro, também podemos interpretar um grafo dirigido como uma "rede de fluxo" e usá-lo para responder a perguntas sobre fluxos de materiais. A teoria de fluxo máximo pode ser usada para responder perguntas do tipo, qual o número máximo de carros uma rodovia pode sustentar, qual a corrente elétrica máxima que um circuito condutor pode suportar, ou ainda qual o volume de água que passa por uma rede de escoamento.

Rede de fluxo máximo

Dada uma fonte s de entrada infinita s e um terminal t, qual a capacidade máxima de fluxo pode atravessar a rede conectada dado que as arestas possuem uma certa capacidade?

- ✓ Um grafo de fluxo é um grafo onde cada aresta (ou arco) possui uma certa capacidade no qual recebe uma certa quantidade de fluxo.

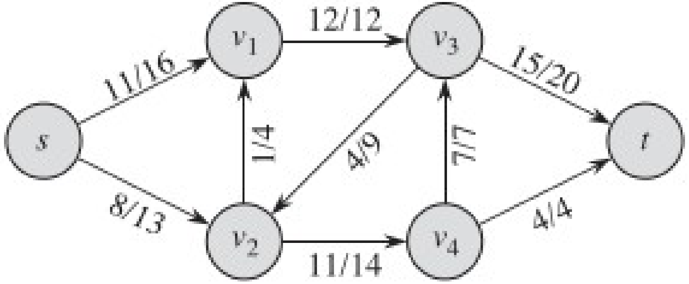


(a)

Propriedades

Os atributos de capacidade e fluxo são representados pela fração adjacente em cada aresta.

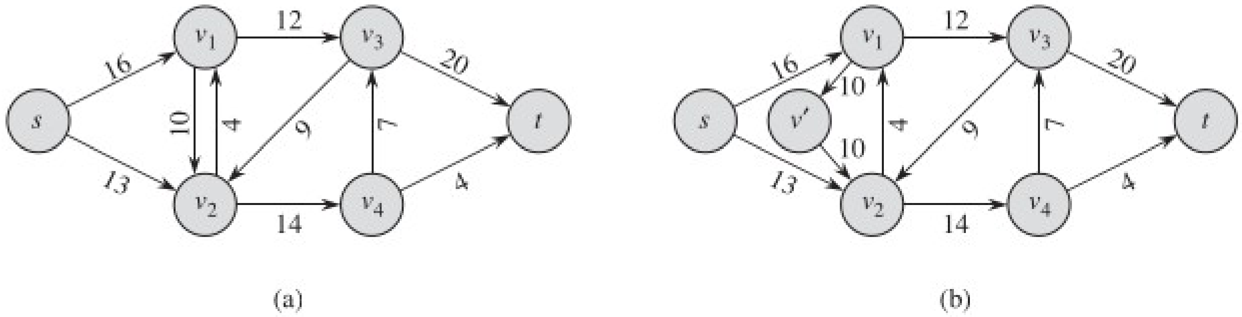
- ✓ **Restrição da capacidade**
  - 0 <= f(u,v) <= c(u,v), para todo (u,v) em V
- ✓ **Conservação do fluxo**
  - sum(f(v,u) = sum(f(u,v)), para u em V - {s,t}
- ✓ Quando não há arestas entre u e v, f(u,v) = 0



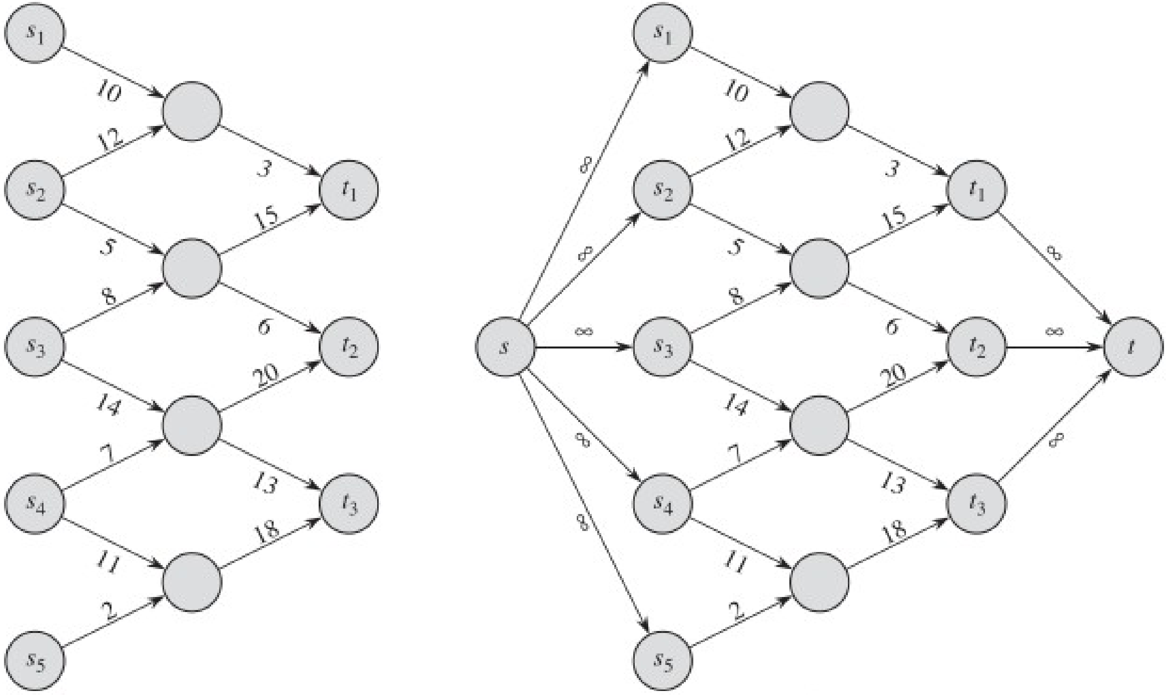
(b)

Rede de fluxo máximo

Não podemos modelar um problema de fluxo máximo com arestas antiparalelas. Para contornar essa situação, temos que transformar o grafo com arestas antiparelas em um grafo equivalente.



Redes de fluxos com múltiplas fontes e terminais



(a) (b)

O método Ford-Fulkerson

O método repetidamente encontra os caminhos aumentadores através da rede residual e aumenta o fluxo até que nenhum caminhos aumentadores possa ser encontrado.

- Um caminho aumentador é um caminho de arestas em um grafo residual com uma capacidade de uso maior que zero da fonte s até o terminal t

FORD-FULKERSON-METHOD(G, s, t)

inicializar fluxo f como 0
while existir um caminho aumentador p na rede residual G_f
    aumentar fluxo f ao longo de p
return f

Rede residual

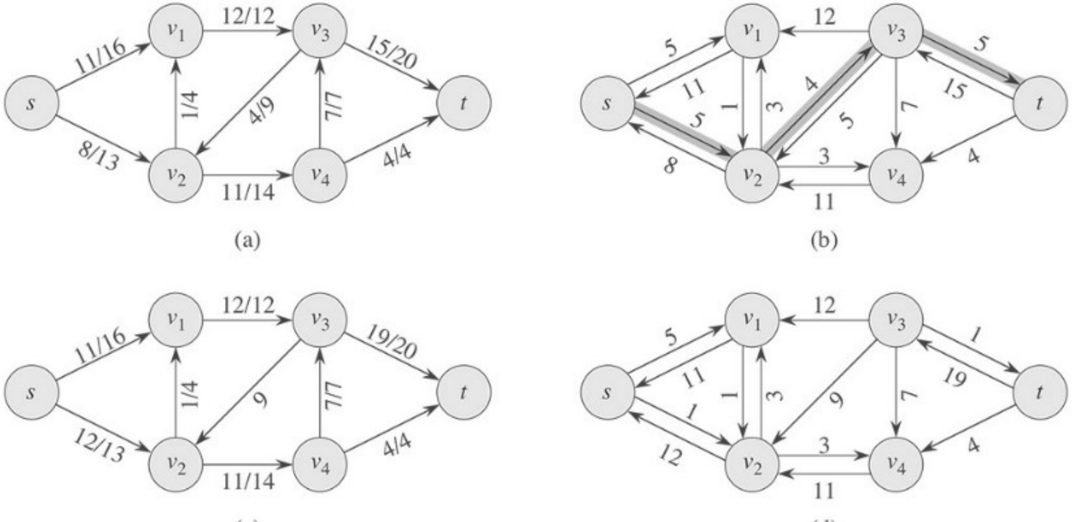
A rede residual é um grafo Gf onde as capacidades das arestas representam como transportar o fluxo das arestas em G.

- Capacidade residual é a diferença entre a capacidade e o fluxo em uma aresta de G.
- Se não há capacidade residual a aresta não pertence a Gf.
- Pode conter arestas que não estão em G
- Arestas invertidas podem ser necessárias ao tentar aumentar o fluxo total

$$c_f(u,v) = \begin{cases} c(u,v) - f(u,v) & \text{se } (u,v) \in E\ , \\ f(v,u) & \text{se } (v,u) \in E\ , \\ 0 & \text{caso contrário}\ . \end{cases}$$

Caminho Aumentador

- Aumentar o fluxo significa atualizar os valores de fluxo das arestas ao longo do caminho aumentador sem infringir a restrição de capacidade.
- O valor do incremento corresponde ao valor do gargalo.



(a) (b) (c) (d)

$$c_f(p) = \min\{c_f(u,v) : (u,v) \text{ is on } p\}$$

$$f_p(u,v) = \begin{cases} c_f(p) & \text{if } (u,v) \text{ is on } p\,, \\ 0 & \text{otherwise}\,. \end{cases}$$

Then, f_p is a flow in G_f with value $|f_p| = c_f(p) > 0$.

Pseudo código

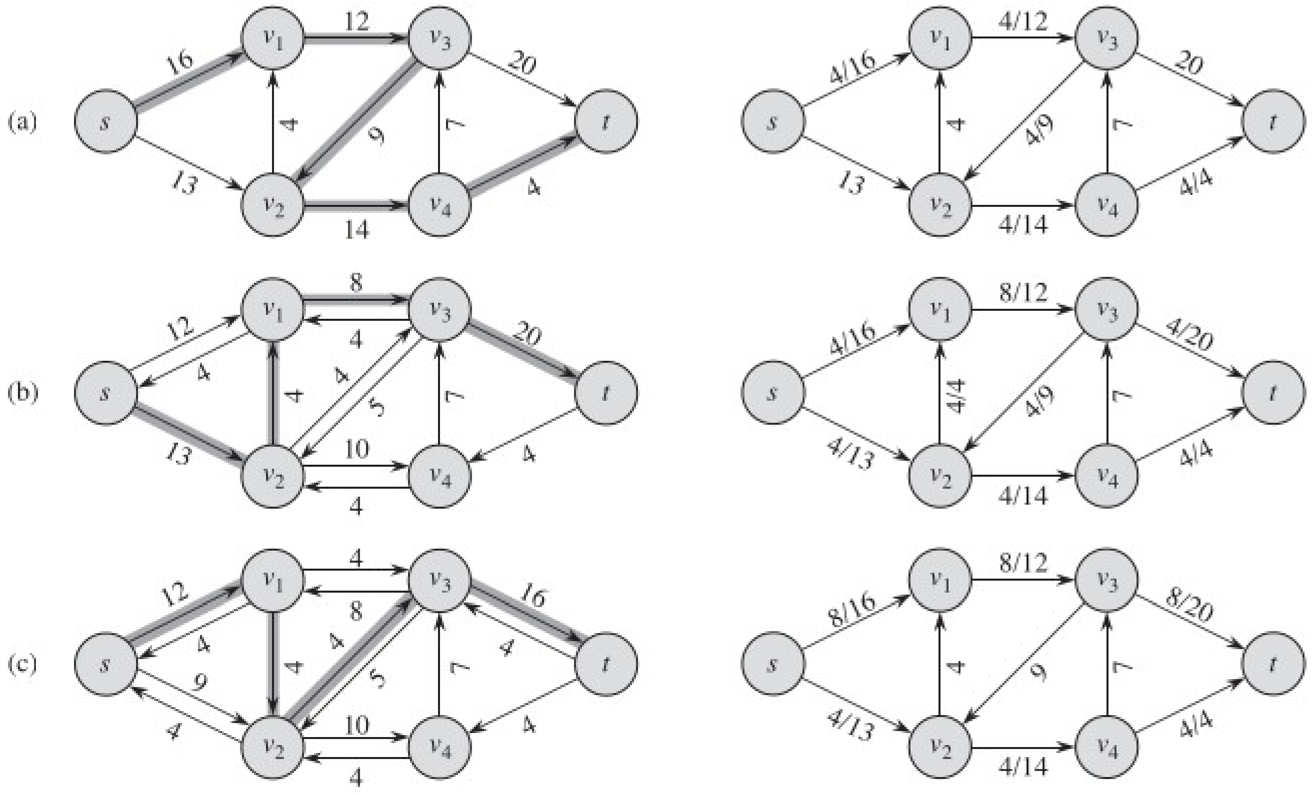
É importante destacar que o método Ford-Fulkerson não especifica um algoritmo para encontrar os caminhos aumentadores. Nesse exemplo vamos consideraramos a busca em profundidade.

FORD-FULKERSON(G, s, t)

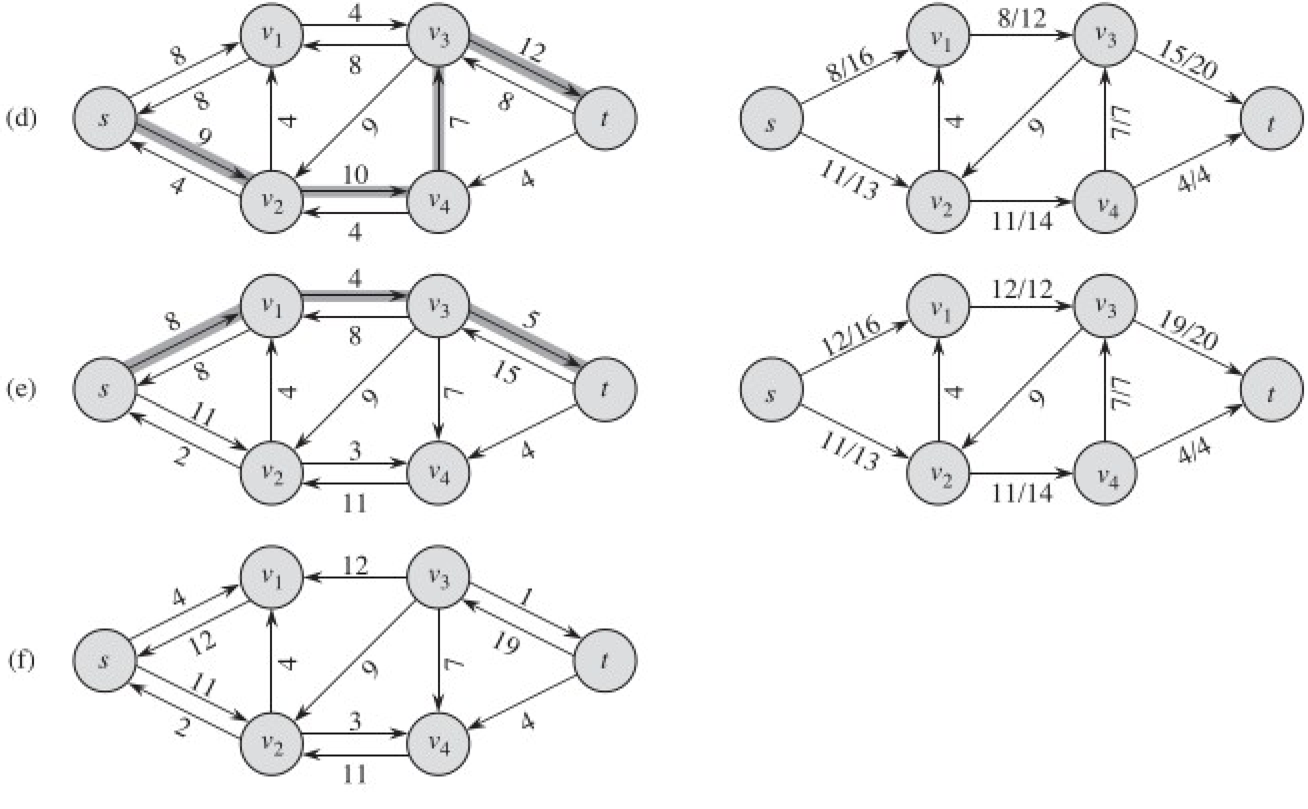
for cada aresta $(u, v) \in G.E$
    $(u, v).f = 0$
while existir um caminho p de s a t na rede residual G_f
    $c_f(p) = \min\{c_f(u, v) : (u, v)$ está em $p\}$
    for cada aresta (u, v) em p
        if $(u,v) \in E$
            $(u, v).f = (u, v).f + c_f(p)$
        else $(v, u).f = (v, u).f - c_f(p)$

$$|f \uparrow f_p| = |f| + |f_p| > |f|.$$

Execução do Ford-Fulkerson



Execução do Ford-Fulkerson



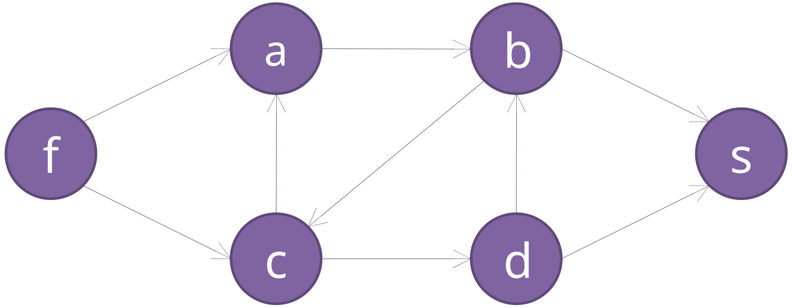
Teorema de fluxo máximo/caminho mínimo

- O corte mínimo de uma rede é um corte cuja a capacidade é mínima em relação a todos os cortes da rede.
- O teorema do fluxo máximo/caminho mínimo garante que há um fluxo máximo apenas quando não existir nenhum caminho aumentador na rede residual.

O algoritmo de Edmonds-Karp

Usa a busca em largura como algoritmo para encontrar o caminho aumentador que proporciona um menor tempo de execução na ordem de O(VE2), bem melhor que a busca em profundidade.

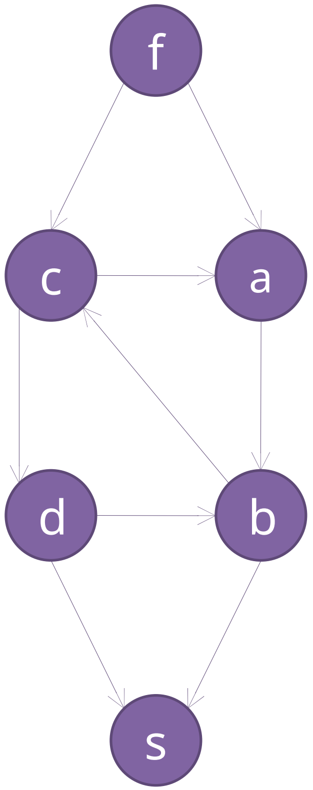
- Encontra o menor caminho aumentador em termos de número de arestas.
- A probabilidade de encontrar gargalos com valores baixo em um caminho longo é maior.
- Um vértice só pode ser explorado se a sua aresta tiver um valor maior que zero.



O algoritmo push-relabel

Funciona de uma maneira mais localizada que o método Ford-Fulkerson. Esses algoritmos agem em um vértice por vez, examinando somente os vizinhos do vértice na rede residual

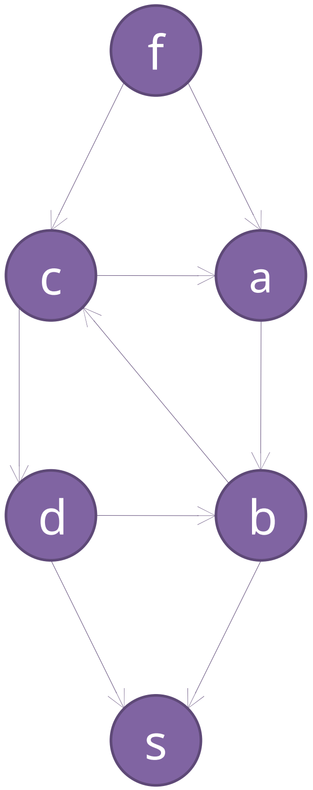
- Possui o tempo de execução mais rápido.
- Não mantém a propriedade de conservação de fluxo.
- Calcula um pré-fluxo que pode transbordar dentro de um vértice.



O algoritmo push-relabel

Fazendo analogia como uma rede de tubos de água, onde as arestas são os tubos e os vértices são as junções. Após a inicialização, o algoritmo aplica a operação de empurrão ou remarcar até que não mais seja possível.

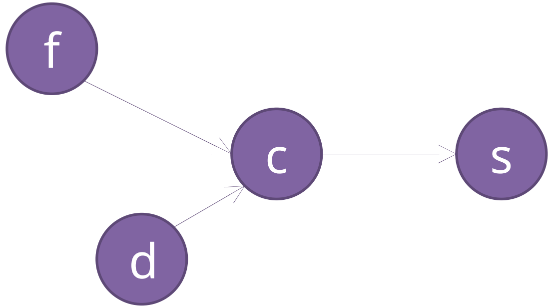
- Empurra(push) o fluxo para baixo
- Remarca os vértices com capacidade excedente.
- O vértice fonte possui altura N e o sorvedouro s possui altura 0.
- Os outros vértices iniciam altura com 0 e aumentam a cada iteração.
- Há um transbordamento quando o excesso de fluxo é maior que 0.



O Algoritmo relabel-to-front

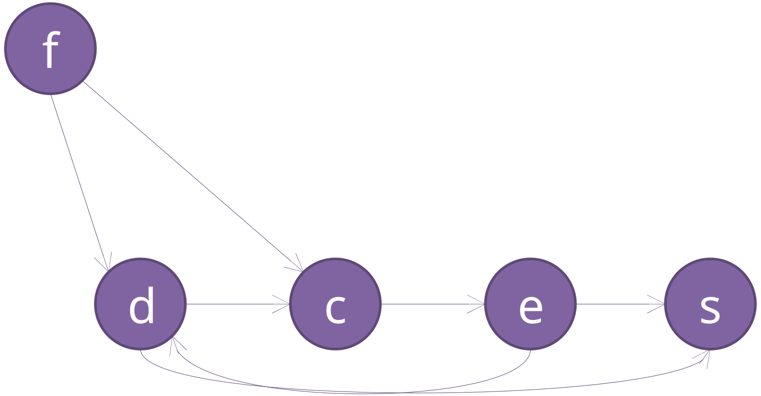
Escolhendo a ordem com cuidado e administrando eficientemente a estrutura de dados da rede, podemos resolver o problema de fluxo máximo mais rapidamente que o limite *O(V2E)*

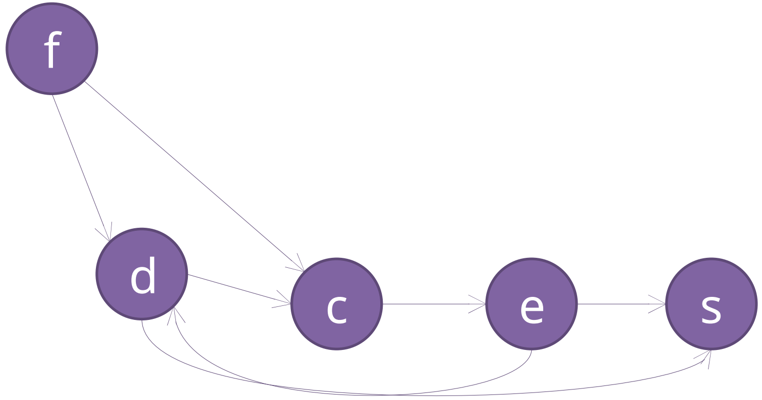
- Mantém uma lista de vértices da rede.
- Varre a lista repetidamente e Descarrega o vértice que está transbordando.
- Sempre que o vértice é desmarcado, ele é colocado na frente da lista.
- Depende da noção de arestas admissíveis.



O Algoritmo relabel-to-front

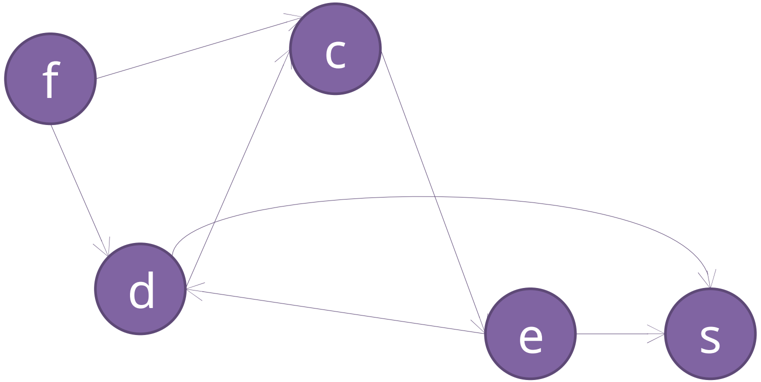
A lista de vértices é composta de {d,c,e}. Cada vértice tem seu vizinho mostrado na tabela. O vértice escolhido para iniciar é d.

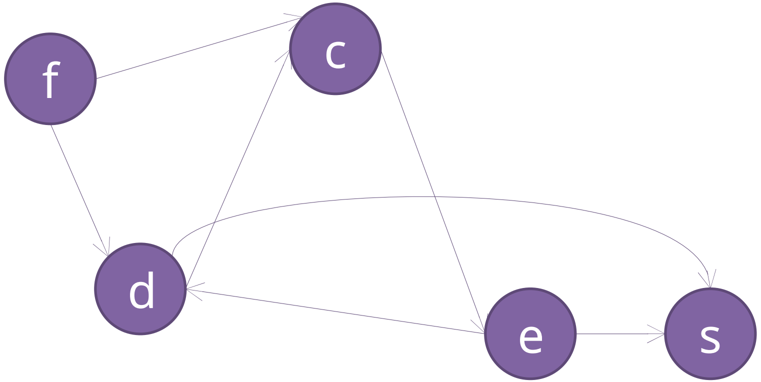




O Algoritmo relabel-to-front

O algoritmo irá precisar de duas listas: uma para os vértices, e outra para seus vizinhos. Possui dois ponteiros: um para o vértice atual e outra para o próximo. E ainda, precisa de uma variável para armazenar a altura antiga do vértice.





Revisão

Atenção, chegou a hora da revisão.