Teoria dos Grafos - BCC 204

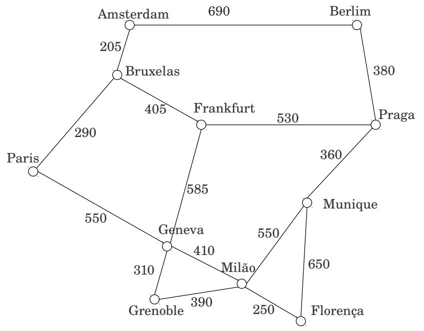
Haroldo Gambini Santos

Universidade Federal de Ouro Preto - UFOP

3 de abril de 2011

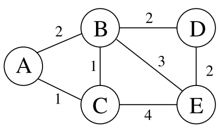
Notas

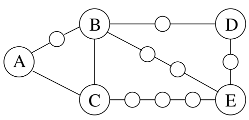
Grafos com Pesos - Computando caminhos mínimos



Notas

Usando BFS





Notas

Algoritmo do Alarme

- Coloque o alarme do nó s para o tempo 0
- Repita enquanto houverem alarmes:
  - pegue o nó u com alarme no menor tempo t
  - para cada vizinho v de u faça:
    - se não houver alarme para v ajuste o alarme de v para $t + dist(u, v)$
    - se houver alarme para v e for maior que o tempo $t + dist(u, v)$, ajuste o mesmo para esse tempo menor

Notas

Algoritmo do Alarme

- Computa caminhos mínimos para qualquer grafo com pesos positivos
- Como implementar sistema de alarmes ?

Notas

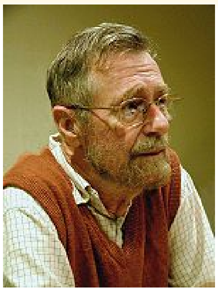
Sistema de Alarmes

Fila de Prioridades - operações que necessitamos

- `inicializa`$(heap, alarmes)$
- `diminui`$(heap, elemento, alarmeMenor)$
- `pegaMenor`$(heap)$

Notas

O Algoritmo de Dijkstra



Edsger W. Dijkstra 1930 - 2002

1959 - Algoritmo de Dijkstra para Caminhos Mínimos

Notas

Algoritmo de Dijkstra

função `dijkstra`(V, A, d, s)
Entrada: conjunto V de vértices e A de arestas, distâncias $d_{(v,u)}$ e origem s.
Saída: $dist$ e $prev$: vetores com distâncias e antecessores nos caminhos.

para todo $u \in V$ **faça**
  $dist_u = \infty$;
  $prev_u =$ `nulo`;
$dist_s = 0$;
`inicializa`$(heap, dist)$;
enquanto `tamanho`$(heap) > 0$ **faça**
  $v =$ `pegaMenor`$(heap)$;
  para todo $u : (v, u) \in A$ **faça**
    se $dist_u > dist_v + d_{(v,u)}$ **então**
      $dist_u = dist_v + d_{(v,u)}$;
      $prev_u = v$;
      `diminui`$(heap, u, dist_u)$;

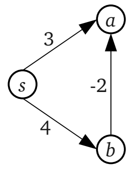
Notas

Complexidade

- Estruturalmente idêntico ao BFS
- Operação mais difícil em fila: sem as operações em tempo constante `pegaPrimeiroDaFila` e `enfileira`
- Operações:
  - $|V|$ operações de `pegaMenor`
  - $|V| + |E|$ operações de `insere/diminui`
  - $O(|V| \times custoPegaMenor + |E| \times custoDiminui)$
  - *Heap* binária: $O((|E| + |V|) \times \log(|V|))$

Notas

Dijkstra - Limitação



- no algoritmo, qualquer caminho de s para outro vértice v deve passar apenas por **vértices mais próximos** de s;
- no exemplo, o caminho mais curto entre s e a passa por b, que é mais distante do que a !
- observação importante:
  - em um grafo com $|V|$ vértices, qualquer caminho entre s e v pode ter no máximo $|V| - 1$ arcos.

Notas

Caminhos Mínimos - Aplicações

Além das aplicações triviais, que são o encontro de caminhos com menor distância/tempo em um grafo, podemos citar a importância de problemas de caminhos mínimos:

- Como sub-rotina em problemas diversos:
  - Problemas de Projeto de Redes
  - Problemas de Multi-Fluxo
  - Problemas de Fluxo de Custo Mínimo
  - ...

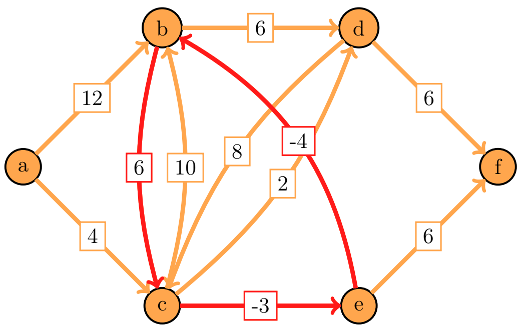
Notas

O Algoritmo de Bellman-Ford

função `bford`(V, A, d, s)
Entrada: conjunto V de vértices e A de arestas, distâncias $d_{(v,u)}$ e origem s.
Saída: $dist$ e $prev$: vetores com distâncias e antecessores nos caminhos.
para todo $u \in V$ **faça**
    $dist_u = \infty$;
    $prev_u =$ `nulo`;
$dist_s = 0$;
para $i \leftarrow 1$ **até** $|V| - 1$ **faça**
    para todo $(v, u) \in A$ **faça**
        se $dist_v + d_{(v,u)} < dist_u$ **então**
            $dist_u \leftarrow dist_v + d_{(v,u)}$;
            $prev_u \leftarrow v$;

Notas

Ciclos de Custo Negativo



Notas

Ciclos de Custo Negativo

Bellman-Ford - Detecção

- em caminhos sem ciclos, o caminho mais longo consiste em $|V| - 1$ arestas, ou iterações no laço principal do algoritmo;
- se na iteração $|V|$ do algoritmo alguma atualização de distâncias for feita é detectado o ciclo.

Notas

Caminhos Mínimos entre Todos os Pares

$dist(i, j, k)$: o comprimento do caminho mais curto entre i e j tal que apenas os nós $\{1, \ldots, k\}$ podem ser usados como intermediários.

Relação de recorrência

$$dist(i, j, k) = \min\{dist(i, k, k-1) + dist(k, j, k-1), dist(i, j, k-1)\}$$

Notas

O Algoritmo de Floyd-Warshall

função `fwarshall`(V, A, d)

Entrada: conjunto V de vértices e A de arestas, distâncias $d_{(v,u)}$.

Saída: Matrizes *dist* e *next* contendo distâncias e sucessores nos caminhos mínimos entre pares de nós.

$dist(u,v) = \infty, pred(u,v) = nulo \quad \forall (u,v) \notin A$;

$dist(u,v) = d_{(u,v)}, pred(u,v) = u \quad \forall (u,v) \in A$;

para $k = 1$ **até** $|V|$ **faça**

  para $i = 1$ **até** $|V|$ **faça**

    para $j = 1$ **até** $|V|$ **faça**

      se $dist(i,k) + dist(k,j) < dist(i,j)$ **então**

        $dist(i,j) \leftarrow dist(i,k) + dist(k,j)$;

        $pred(i,j) \leftarrow pred(k,j)$;

Notas

Notas

Notas