Caminho mais curto em Grafos

SCC 503 - Alg. Estrutura de Dados II

Você sabe o que é um algoritmo guloso?

- Tradicionalmente, o cálculo do caminho mais curto de um vértice aos demais em um grafo utiliza uma estratégia gulosa.
- É difícil definir com precisão o que é um algoritmo guloso:
    a. tal algoritmo constrói uma solução em passos pequenos, escolhendo uma decisão, em cada passo, meio que de forma “míope”, que otimize algum critério subjacente.
    b. Um algoritmo guloso faz a escolha ótima localmente, a cada passo, na esperança de finalmente chegar a solução ótima globalmente.
- Para que um problema aceite uma solução gulosa, este deve ter as seguintes propriedades
    a. Ter sub-estruturas ótimas -> solução ótima global contém solução ótima para os sub-problemas
    b. ter a propriedade gulosa -> Se fizermos a escolha que parece ser a melhor naquele momento, e prosseguimos nos sub-problemas remanescentes, chegaremos a solução ótima. Nunca será preciso reconsiderar escolhas prévias (difícil de provar…)

Um exemplo de problema que aceita alg. guloso

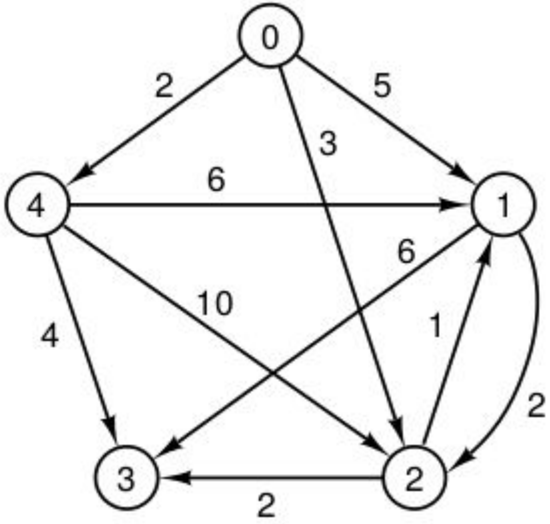
- Troca de moedas
  - Dada uma quantia de *V* centavos e uma lista de *n* moedas disponíveis, qual é o número mínimo de moedas que precisamos para representar V? (o nro de moedas é ilimitado)
  - Se n = 4, coinValue = {25, 10, 5, 1}.
  - como representar V = 42 ?
  - Estratégia Greedy:
    - selecione a maior moeda, que não seja maior que o troco remanescente!
    - 42 - 25 = 17 -> 17-10 = 7 -> 7-5 = 2 -> 2 -1 = 1 -> 1 - 1 = 0. terminou
      - O problema tem sub-estruturas ótimas
        - 42 = 25+10+5+1+1 (problema global)
        - 17 = 10+5+1+1 (sub problema tb é ótimo)
        - 7 = 5 + 1 + 1 (sub problema tb ótimo), etc…
      - tem propriedade gulosa:
        - para ESTE conjunto de moedas coinValue, nenhuma outra estratégia trará solução melhor.
  - Seja coinValue = {4,3,1} e V = 6. Como resolver isso????

Caminho mais curto - Shortest Path - em grafos

- Existem 2 categorias de “problemas” de caminhos mais curtos
  - Single-source shortest Path (SSSP)
    - Dado um grafo G ponderado e um vértice s qualquer, encontre o menor caminho de s para TODOS os demais vértices do grafo
    - o grafo não pode ter ciclos negativos !
  - All-pairs Shortest Path
    - Dado um grafo conexo e ponderado *G*, encontre o menor caminho entre os vértices *s* e *d*, para <u>TODOS</u> vértices *s* e *d* em G.
    - pode haver ciclos negativos no grafo.

Exemplo de um grafo

Para esta aula, vamos usar o seguinte grafo conexo e dirigido

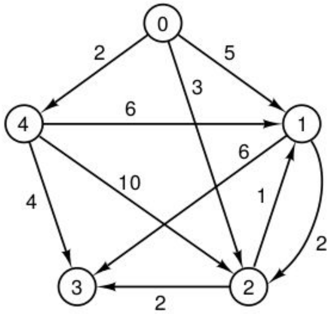


SSSP em grafos sem pesos

- Responda sem titubear (suponha que o grafo anterior não tenha pesos)
  - É preciso explicar o algoritmo que calcula a menor distância de um vértice s em um grafo para todos os demais ou já vimos isso em aulas passadas ????
- Vamos pensar que além de calcular o caminho mais curto de v até os demais vértices, nós queiramos “imprimir” este caminho… o que poderíamos fazer?

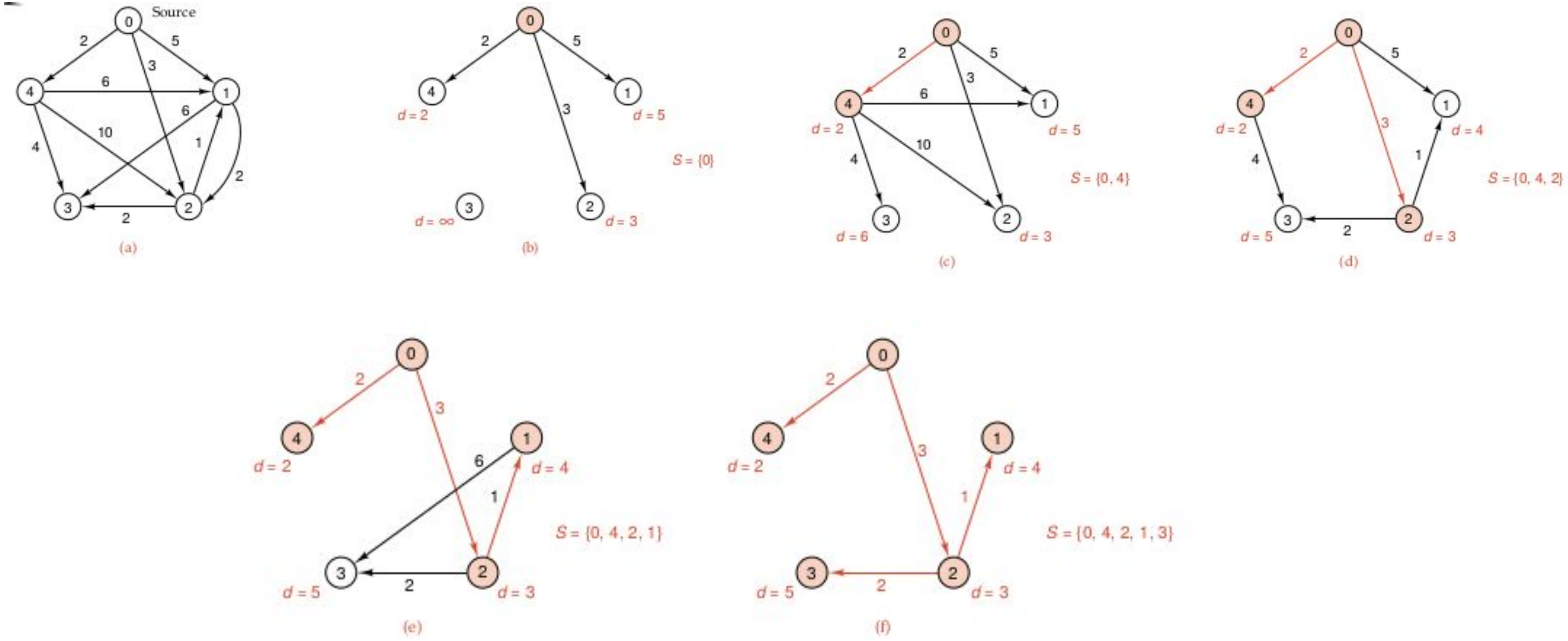
SSSP em grafos com peso - Alg. de Edsger Dijkstra

Claramente o menor caminho entre 0 e 1 é 4 com a rota 0 -> 2 -> 1.



1. Inicialmente, a partir do vértice **s** (=0) iniciamos o vetor de distância com o que temos (os vértices não alcançáveis tem distância infinita
2. Seja um conjunto S que contenha os vértices para os quais já calculamos a distância mínima de **s** até ele
    - ora, ora.. inicialmente S = {**s**}, certo? Este será colorido.
3. Temos um vetor dist que armazena a distância de **s** para todo o vértice **v**. Se **v** está em S, então a dist[**v**] já tem a menor distância. Se **v** não está em S, então dist[v] dá a distância de **s** a algum vértice **w** em S + o peso da aresta (w->v)
4. para adicionar um novo vértice à S, aplicamos o critério guloso.
    - escolha um vértice v com a menor distância em dist, tal que v ainda não esteja em S.

Veja o algoritmo guloso de Dijkstra em ação



Prova de que esta estratégia gulosa funciona

- Devemos provar que para cada vértice v, a distância gravada em dist é a menor distância de s (0) até v.
- Suponha que houvesse uma trilha mais curta de s até v, como na figura ao lado.
- Esta trilha abdicaria do conjunto S e usaria um vértice x para chegar a v…..
- Mas se esta trilha é mais curta do que a trilha colorida até v, então este segmento inicial de 0 a x seria também mais curto. E pelo critério guloso adotado, x já estaria no conjunto S pois seria escolhido ANTES de v, pois dist[x] < dist[v] !!!!!
- Entendeu?????