Problema 1

Exercício (CLRS 24.3-6). Seja $G = (V, E)$ um grafo orientado em que a cada aresta $(u, v) \in E$ está associado um número real $r(u, v)$, onde $0 \leq r(u, v) \leq 1$, que representa a confiabilidade do canal de comunicação de u a v. Interprete $r(u, v)$ como a probabilidade do canal (u, v) **não** falhar e suponha que as probabilidades são independentes.

Dados dois vértices s e t em G, mostre como encontrar eficientemente um st-caminho de confiabilidade máxima (maior probabilidade de nenhuma aresta do caminho falhar).

Solução

- Um caminho $P = (s = v_1, v_2, \ldots, v_k = t)$ de confiabilidade máxima é um que maximiza $r(P) = r(v_1, v_2) \times r(v_2, v_3) \times \cdots \times r(v_{k-1}, v_k)$.
- Considere o grafo ponderado (G, ω) em que $\omega(u, v) = -\log r(u, v)$. Como $0 \leq r(u, v) \leq 1$, $\log r(u, v) \leq 0$ e portanto, $\omega(u, v) \geq 0$.
- O peso de um st-caminho $P = (s = v_1, v_2, \ldots, v_k = t)$ é:

$$
\begin{aligned}
\omega(P) &= \omega(v_1, v_2) + \omega(v_2, v_3) + \cdots + \omega(v_{k-1}, v_k) \\
&= -\log r(v_1, v_2) - \log r(v_2, v_3) - \cdots - \omega(v_{k-1}, v_k) \\
&= -[\log r(v_1, v_2) + \log r(v_2, v_3) + \cdots + \omega(v_{k-1}, v_k)] \\
&= -\log[r(v_1, v_2) \times r(v_2, v_3) \times \cdots \times r(v_{k-1}, v_k)] \\
&= -\log r(P).
\end{aligned}
$$

- Logo, um st-caminho que minimiza $\omega(P)$ é um que maximiza $r(P)$ e vice versa.
- Usando o algoritmo de Dijkstra implementado com heaps, podemos resolver o problema em tempo $O(E \lg V)$. Usando heap de Fibonacci, podemos resolver o problema em tempo $O(E + V \lg V)$

Problema 2

Exercício (CLRS 24.3-8). Seja (G, ω) um grafo orientado sem arestas negativas onde os pesos das arestas são inteiros dentro do intervalo de 0 a W, onde W é uma constante. Mostre como modificar o algoritmo de Dijkstra para resolver o Problema dos Caminhos Mínimos com Mesma Origem em tempo $O(WV + E)$.

Sugestão. Mostre como implementar uma fila de prioridades adequada. Explique sucintamente, mas cuidadosamente como cada operação é implementada.

Observação. (CLRS 24.3-9) É possível resolver o problema em tempo $O((V + E) \lg W)$. (Exercício)

Solução

- Lembre que no algoritmo de Dijkstra os vértices são inseridos na fila de prioridade em ordem crescente de distância da origem s.
- Note que um caminho (simples) pode ter no máximo $|V| - 1$ arestas e portanto, peso no máximo $W(|V| - 1)$. Assim, os possíveis valores de $d[v]$ (estimativas das distâncias) são $0, 1, 2, \ldots, W(|V| - 1)$ e ∞.
- Temos um vetor $L[0 \ldots W(|V| - 1) + 1]$. ($W(|V| - 1) + 1$ representa $+\infty$). Na posição $L[k]$ temos uma lista duplamente ligada contendo todos os vértices v com chave $d[v] = k$.
- INSERT consome tempo $O(1)$ pois simplesmente inserimos o vértice com chave k em $L[k]$.

Solução

- Para fazer o EXTRACT-MIN basta manter um apontador para a primeira lista não-vazia. Se uma lista $L[k]$ ficar vazia, então movemos o apontador para a próxima posição de L que contém uma lista não-vazia. Não precisamos voltar no vetor L, pois os vértices são inseridos na fila em ordem crescente de distância de s. O custo total de todas as chamadas de EXTRACT-MIN é $O(WV)$.
- DECREASE-KEY consome tempo $O(1)$. Para reduzir a chave k para um valor k' de um vértice v, remova v de $L[k]$ e o insira em $L[k']$. Note que é necessário ter para cada vértice um apontador para seu nó correspondente. Como cada lista é duplamente ligada, podemos remover um nó em tempo $O(1)$.

Tempo total: (visto em aula)
$O(V)$ INSERT $+$ $O(V)$ EXTRACT-MIN $+$ $O(E)$ DECREASE-KEY $=$
$O(V + WV + E) = O(WV + E)$.

Problema 3

Exercício (Lista de PL no site). Uma fábrica de aço produz quatro tamanhos de "I beams" (consulte a Wikipedia!): pequeno, médio, grande e extragrande. Esses beams podem ser produzidas por qualquer das máquinas A, B e C. Os comprimentos em metros de I beams que podem ser produzidas nas máquinas em uma hora estão indicadas abaixo.

Beam	A	B	C
pequeno	300	600	800
médio	250	400	700
grande	200	350	600
extragrande	100	200	300

Suponha que as máquinas A, B e C podem ser usadas até 60, 55 e 50 horas por semana e que seus custos de operação (por hora) são R$30, R$50 e R$80, respectivamente. Suponha ainda que 10.000, 8.000, 6.000 e 6.000 metros dos diferentes tipos de I beams são requisitados por semana. Formule o problema de escalonar as máquinas de modo a minimizar o custo total de produção como um programa linear.

Solução

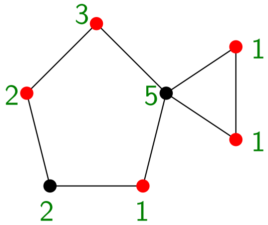
- Variáveis: $x_{ij} =$ quantidade de horas que a máquina j gasta em uma semana de produção do I beam do tipo i, para $i = 1, 2, 3, 4$ (Tipos de I beam: pequeno, médio, grande, extragrande) e $j = A, B, C$ (Máquinas).
- Função objetivo:
$$\min z = 30(x_{1A} + x_{2A} + x_{3A} + x_{4A}) + 50(x_{1B} + x_{2B} + x_{3B} + x_{4B}) + 80(x_{1C} + x_{2C} + x_{3C} + x_{4C})$$
- Restrições de limite de tempo por semana de cada máquina
$$x_{1A} + x_{2A} + x_{3A} + x_{4A} \leq 60$$
$$x_{1B} + x_{2B} + x_{3B} + x_{4B} \leq 55$$
$$x_{1C} + x_{2C} + x_{3C} + x_{4C} \leq 50$$

Solução

- Restrições de demanda semanal de cada tipo de I beam
  $300x_{1A} + 600x_{1B} + 800x_{1C} \geq 10.000$
  $250x_{2A} + 400x_{2B} + 700x_{2C} \geq 8.000$
  $200x_{3A} + 350x_{3B} + 600x_{3C} \geq 6.000$
  $100x_{4A} + 200x_{4B} + 300x_{4C} \geq 6.000$
- Restrições de não-negatividade:
  $x_{ij} \geq 0$ para $i = 1, 2, 3, 4$ e $j = A, B, C$

Problema 4

Exercício. Uma cobertura (de vértices) de um grafo $G = (V, E)$ é um subconjunto $K \subseteq V$ tal que para toda aresta $uv \in E$, $u \in K$ ou $v \in K$.
Suponha que cada vértice $v \in V$ tem associado um número real $\omega(v)$, o peso de v. Uma cobertura de peso mínimo de G é uma cobertura K que minimiza a soma dos pesos dos vértices em K.
Formule o problema de encontrar uma cobertura mínima de G como um problema de programação linear inteira.



Solução

- Variáveis: x_v para $v \in V$
  ($x_v = 1$ significa que v faz parte da cobertura, caso contrário, $x_v = 0$).
- Função objetivo:
  $\min z = \sum_{v \in V} x_v \omega(v)$
- Restrições de cobertura:
  $x_u + x_v \geq 1$ para toda aresta $uv \in E$
- Não-negatividade: $x_v \geq 0$ para todo vértice $v \in V$
- Integralidade: $x_v \in \mathbb{Z}$ para todo $v \in V$ (é possível mostrar que para uma solução ótima inteira x, $x_v \in \{0, 1\}$ para todo $v \in V$)
- **Alternativa:** poderia trocar as duas últimas restrições por $x_v \in \{0, 1\}$ para todo $v \in V$