MO417 - Ata do exercício

Fábio Augusto Faria - RA079734 - 09 de junho de 2010

Exercício 24.3-6

Enunciado

Seja $G = (V, E)$ um grafo orientado ponderado com função peso $w : E \rightarrow 0, 1, ..., W$ para algum inteiro não negativo W. Modifique o algoritmo de Dijkstra para calcular os caminhos mais curtos a partir de um vértice de origem s dado o tempo O($WV + E$)

Introdução

O colega Pedro Feijão iniciou a explicação preenchendo a Tabela 1 começando pelo Vetor, Heap Binário e por última a Nova Estrutura. Esta Nova Estrutura foi preenchida mantendo os custos unitários das operações INSERT e RELAX no Vetor (O(1)) e notou-se que o custo unitário da operação EXTRACT-MIN deveria ser O(W) para que o objetivo fosse alcançado (O($WV + E$)).

		Custo Unitário das estruturas		
Operações	Quantidade	Vetor	Heap Binário	Nova Estrutura
INSERT	$\|V\|$	O(1)	O(lgV)	O(1)
RELAX	$\|E\|$	O(1)	O(lgV)	O(1)
EXTRACT-MIN	$\|V\|$	O(V)	O(lgV)	**O(W)**
Custo Total	-	O(V^2)	O(($V + E$) $*$ lgV)	**O($WV + E$)**

Tabela 1: Tempo de execução do algoritmo de Dijkstra utilizando Vetor, Heap Binário e a Nova Estrutura proposta para resolver este exercício.

Na próxima seção será mostrada em detalhes a Nova Estrutura proposta para resolver o exercício.

Resolução

A Nova Estrutura proposta é um vetor (VET) com tamanho $(V - 1) \times W$, que representa todos os valores de distâncias possíveis que um vértice pode ter. Este vetor substituirá a fila de prioridade do algoritmo de Dijkstra original.

Em cada posição, $i = \{0, 1, 2, ..., (V - 1) \times W\}$, de VET existirá um ponteiro para uma lista de vértices (vide Figura 1) que terão o valor de distância igual a i. Após a inicialização do algoritmo de Dijkstra (INITIALIZE-SINGLE-SOURCE) apenas o vértice **"s"**, que é o vértice inicial, se encontrará em VET (VET[0]), pois sua distância é igual a 0 e os demais vértices estão em uma lista duplamente encadeada com distâncias iguais a ∞ (vide Figura 2).

1. **INSERT**: Como os vértices serão inseridos em uma lista duplamente encadeada o custo da operação será O(1).

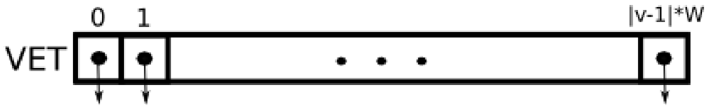

Figura 1: Estrutura proposta para resolver o exercício.

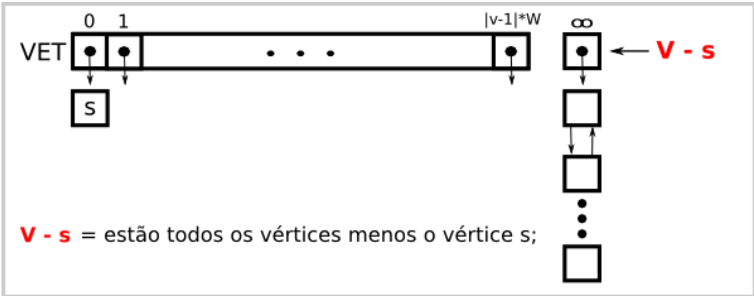

Figura 2: Depois de chamar a função INITIALIZE-SINGLE-SOURCE.

2. **RELAX**: O acesso direto aos vértices, inserção e remoção no início de listas duplamente encadeadas possibilitam que o custo da operação RELAX seja O(1). Após o relaxamento, o valor da distância do vértice será modificado, o vértice será removido da sua posição atual e inserido em uma nova posição em VET.

3. **EXTRACT-MIN**: Nesta operação, ao retirar o vértice de menor distância (VET[i]) a sua posição i deve ser guardada, pois a posição do próximo vértice a ser retirado será sempre maior ou igual (>=) a posição i. Essa afirmação é verdadeira, pois cada posição em VET representa o valor da distância de um vértice à **"s"**, e sabemos que a distância do próximo vértice a ser retirado de VET será sempre maior ou igual à do vértice anterior. Não ter que percorrer VET sempre da posição $i = 0$, a cada chamada EXTRACT-MIN, reflete no custo igual a O(W).

Conclusão

O algoritmo de Dijkstra executa $|V|$ operações de EXTRACT-MIN com custo O(W), totalizando um custo de O(WV) e $|E|$ operações de RELAX com custo O(1), totalizando um custo de O(1×E). Como foi visto na Tabela 1 essa implementação terá custo total O($WV + E$).