Trabalho Prático - Diâmetro (Módulo 2)

1 Modelagem do Problema

Seja $G = (V, E)$ um grafo conexo ponderado, tal que o peso de uma aresta $(i, j) \in E$ é dado por $w(i, j)$, para determinar o diâmetro de G primeiro faz-se necessário calcular o caminho mínimo $\delta(i, j)$ para todo $(i, j) \in E$. Para tanto, foi utilizado o algoritmo de `Floyd-Warshall` ligeiramente modificado. O algoritmo original contém um único laço principal, que vai de $k = 1$ até n, onde $n = |V|$. A cada iteração, o algoritmo computa a distância $d_{ij}^{(k)}$ entre i e j de tal forma que os vértices intermediários do caminho estejam em $\{1, \dots, k\}$.

Note que, quando $k = n$, todos os vértices em V podem ser intermediários. Isto é, a última iteração é responsável por calcular, de fato, o caminho mínimo entre cada par de vértices (ou o caminho já foi calculado, ou será atualizado uma última vez). Sendo assim, é possível separar o laço principal em dois laços, o primeiro de $k = 1$ até $n - 1$, e o segundo para $k = n$. Com isso, o segundo laço pode ser utilizado para calcular o diâmetro do grafo: $max\{ \ d_{ij}^{(n)} \mid (i, j) \in E \ \}$.

Além do diâmetro dado pelo maior valor $\delta(i, j)$ em G, deseja-se também encontrar o caminho $i \rightsquigarrow j$. Para tanto, podemos definir uma matriz de predecessores $\Pi = (\pi_{ij})$ ($\Pi = P$ na implementação), sendo $\Pi^{(n)} = \Pi$. Um elemento π_{ij} representa o predecessor do vértice j no caminho de i até j, e é definido de acordo com as equações abaixo. Para determinar o caminho $i \rightsquigarrow j$, basta reconstruí-lo (e.g. com uma *pilha*) a partir de j, utilizando o predecessor π_{ij} até que este seja nulo.

$$\pi_{ij}^{(0)} = \begin{cases} NULO & \text{se } i = j \text{ } ou \text{ } w(i,j) = \infty \\ i & \text{caso contrário} \end{cases} \qquad \pi_{ij}^{(k)} = \begin{cases} \pi_{ij}^{(k-1)} & \text{se } d_{ij}^{(k-1)} \leq d_{ik}^{(k-1)} + d_{kj}^{(k-1)} \\ \pi_{kj}^{(k-1)} & \text{caso contrário} \end{cases}$$

2 Análise de Complexidade

Tempo. A construção do grafo $G = (V, E)$ é feita em duas etapas, sendo no total $O(n+m)$. Na primeira, inicializa-se cada linha da matriz de adjacência, definindo $w(i, i) = 0$. A seguir, tem-se a inserção das m arestas. A implementação do algoritmo de `Floyd-Warshall` descrito acima é composta por três laços. O primeiro, de $i = 0$ até $n - 1$ e $j = 0$ até $n - 1$, refere-se a inicialização das matrizes D e P (Π) em tempo $O(n^2)$. O segundo e o terceiro laço, juntos, correspondem ao laço principal do `Floyd-Warshall` original. Sendo assim, a complexidade final é $O(n^3)$. Por fim, tem-se o tempo gasto para reconstruir e apresentar o caminho entre os *vértices diametrais*. Nessa implementação, utilizou-se uma pilha para inserir os vértices do caminho de trás para frente, percorrendo os predecessores de cada um. Tendo em vista que este é um caminho mínimo, a complexidade da reconstrução e apresentação do caminho é $O(n)$. Portanto, a complexidade total é $O(n + m + n^3 + n) = O(n^3)$, posto que $m = O(n^2)$.

Espaço. O grafo é representado por uma matriz de adjacência, onde cada posição ij corresponde ao peso da aresta $(i, j) \in E$. Além disso, as matrizes D e P também são matrizes quadradas de tamanho $n \times n$. Por fim, o caminho mínimo referente ao diâmetro contém no máximo n vértices. Logo, o espaço total utilizado é $O(3n^2 + n) = O(n^2)$.