# Análise de Síntese e Algorítmos

## Relatório 2ºProjeto

### Grupo 71

Neste trabalho considerou-se o mapa de uma cidade, onde existem pontos, onde as pessoas se podem concentrar, e ligações entre esses pontos. A preocupação principal é garantir que não há comunicação, entre um conjunto de pontos criticos. Isto é, pretende-se que pelo menos um conjunto não vazio de pontos críticos fique isolado dos restantes, minimizando o número de caminhos a bloquear.

O objectivo deste projeto consistia em modelar o problema, utilizando estrutras de dados e algoritmos adequados, por forma a evitar que seja iniciado um motim.

Inicialmente foi necessário representar o mapa da cidade em memória (os seus diferentes pontos e os caminhos entre esses pontos). Para tal foi utilizado um grafo direcionado e não pesado [o peso é sempre igual a 1 (um)]. Um grafo permite representar de forma eficiente cada um dos pontos (representado por um vértice) e os caminhos entre esses pontos (representado por um arco). Apesar do grafo ser direcionado, sempre que é criada uma ligação de um ponto A para um ponto B, é criada também de B para A, pois no contexto do problema os caminhos têm duas direções. Optou-se por utilizador um grafo direcionado, para ser possível impletementar o algoritmo Ford-Fulkerson. Quanto ao peso dos arcos, este tem sempre o valor 1 (um). No contexto do problema não nos é indicada a dimensão desses caminhos (por exemplo, quantas pessoas conseguem passar ao mesmo tempo), nem é relevante, pois apenas queremos saber o número de caminhos que necessitam de ser barrados, e não a capacidade das mesmas.

Segundo o Teorema do Fluxo Máximo-Corte Mínimo, o fluxo máximo num grafo desde um ponto inicial até a um ponto de destino é igual a capacidade mínima, que quando é removida (ligação bloqueada) faz com que passe a ser impossível ir do ponto inicial até ao destino.

O grafo foi guardado em memória através de uma lista de adjacências, pois é a alternativa mais eficaz e eficiente, a nível de memória e tempo, para pesquisar as adjacências de um determinado vértice.

Através de um grafo, é possível pesquisar caminhos entre dois pontos de forma eficaz e eficiente através de diversos algoritmos. Optou-se por recorrer a uma pesquisa BFS (Breadth-First Search). A BFS é percorrida desde o primeiro vértice da pesquisa (source) até o vértice que se pretende atingir (target).

Tendo em conta que apenas é necessário isolar um dos pontos críticos do restante grupo desses mesmos pontos, foram procurados caminhos entre todos os pontos do conjunto critico, e devolvido o caso em que existir o menor número de caminhos. Isto é, se tivermos um conjunto de pontos críticos com 3 pontos (A,B,C), é verificado o número de caminhos entre A => B, A => C e B => C. Apesar do grafo ser direcionado, não faz qualquer sentido pesquisar A=>B e repetir a pesquisa para B => A, pois, tal como explicado anteriormente, quando é criado um caminho de A para B, é também criado um caminho de B para A, logo os caminhos possíveis são iguais. Também foi omitida a pesquisa A=>A, pois o resultado é sempre 0 (zero).

Para calcular o número de caminhos entre dois pontos, optou-se por implementar o algoritmo Ford Fulkerson, que devolve o fluxo máximo entre dois vértices de um grafo. O fluxo máximo de um grafo é, segundo o Teorema da Fluxo Máximo-Corte Mínimo, igual ao número máximo de caminhos existentes entre dois pontos (cada um com a capacidade igual a 1 (um)).

Assim, o número mínimo de ligações a barrar entre dois pontos é igual ao fluxo máximo do grafo entre um ponto origem e um ponto de destino.

A complexidade do algoritmo Ford-Fulkerson é (Ef), e como para cada problema com k pontos criticos, o algoritmo é executado uma vez para cada combinação de 2 (dois) pontos do conjunto, este é executado kC2 vezes. Visto, então a complexidade do algoritmo que resolve cada um dos problemas é:

O( *VC2* ( *Vf* ) ) = O()

com f = fluxo máximo do grafo, E = quantidade de arcos, e V sendo a quantidade de vértices.

Para o resolver h problemas, a complexidade do algoritmo é O()