**Curso: Engenharia da Computação**

**Professor: Douglas Castilho**

**Nome: Maylon Silva Meira e Matheus Moreno**

Nesse primeiro trabalho prático, precisávamos desenvolver um algoritmo utilizando de conhecimentos geométricos, dado um problema onde o programa vai verificar se é possível separar barracas dos homens das barracas das mulheres em uma Conferência Nacional de Computação usando apenas uma parede reta com algumas condições.

Desta forma, criamos esse programa onde lê a entrada do usuário para determinar se é possível separar dois grupos de pontos (homem e mulher) usando uma linha(que seria nossa “parede”). A entrada consiste em vários casos de teste, onde cada caso de teste começa com dois inteiros H e M, representando o número de retângulos para os grupos homem e mulher, respectivamente. O programa termina quando H e M são ambos iguais a 0.

Para cada caso de teste, o programa lê as coordenadas dos retângulos para ambos os grupos. Cada retângulo é representado por dois pontos (x1, y1) e (x2, y2), onde (x1, y1) é o canto inferior esquerdo e (x2, y2) é o canto superior direito. O programa adiciona todos os quatro cantos de cada retângulo às suas respectivas listas (homem e mulher).

Depois de ler a entrada, o programa chama o método possivelSeparacao para determinar se é possível separar os dois grupos usando uma linha. Este método recebe dois argumentos: uma lista de pontos para o grupo homem e uma lista de pontos para o grupo mulher.

O método possivelSeparacao interage sobre todos os pares de pontos de ambos os grupos e cria uma linha entre cada par de pontos. Para cada linha, ele verifica se todos os pontos do grupo homem estão em um lado da linha e todos os pontos do grupo mulher estão do outro lado da linha. Se tal linha existir, então é possível separar os dois grupos e o método retorna true. Caso contrário, se nenhuma linha desse tipo existir, então não é possível separar os dois grupos e o método retorna false.

Depois de chamar o método possivelSeparacao, o programa imprime uma mensagem indicando se é possível ou não separar os dois grupos para o caso de teste atual. Em seguida, ele prossegue para ler a entrada para o próximo caso de teste.

**Análise da complexidade do algoritmo**

O método possivelSeparacao tem uma complexidade de tempo de O(n^2), onde n é o número de elementos nas listas homem e mulher. Isso ocorre porque existem dois loops aninhados neste método, onde o loop externo itera sobre todos os elementos na lista homem, e o loop interno itera sobre todos os elementos na lista mulher para cada iteração do loop externo.

O método main lê dados de um arquivo e preenche as listas homem e mulher. A complexidade de tempo de leitura de dados de um arquivo é O(n), onde n é o número de linhas no arquivo. No entanto, a complexidade de tempo para preencher as listas homem e mulher é O(1) porque adicionar um elemento a uma ArrayList leva um tempo constante.

Portanto, a complexidade de tempo geral do programa é O(n^2) + O(n) + O(1), que simplifica para O(n^2), consideramos apenas o termo de maior ordem ficando apenas **O(n^2)**.