

# CCM0128 - Computação II - 2012

## Exercício Programa 1b - Intervalos Disjuntos

E-mail do monitor: *ademar.mlf@gmail.com*

### O problema

Você acabou de entrar no seu ciclo avançado (uhu!!) e tem que escolher que disciplinas quer cursar no próximo semestre. Como você é um aluno muito interessado, há uma gama muito grande de disciplinas que você quer cursar, mas em muitas delas os horários batem. Não conseguindo decidir pelos tópicos, você decide simplesmente fazer o maior número de disciplinas não conflitantes possível.

Pelo novo regulamento da USP de 2013, cada disciplina é oferecida em apenas um dia da semana, com no mínimo uma hora de duração e deve seguir um horário definido de começo e fim rigorosamente, além disso, esses horários não poderão mais ser quebrados (as aulas não poderão começar ou acabar às 6h30, mas às 6h, por exemplo, é permitido). Além disso, a USP conseguiu montar uma infra-estrutura perfeita que permite aos alunos saírem de uma classe e chegarem a outra instantaneamente.

Seu objetivo neste EP é fazer um programa que, dados os horários de começo e fim das aulas em um certo dia da semana, devolve qual o número máximo de aulas que não colidem.

Você já ouviu falar na idéia de algoritmos gulosos e pensa em resolver o problema usando esse paradigma. Sabendo que você vai precisar de um método de ordenação para usar um algoritmo guloso, você decide implementar o quicksort para ordenar as aulas. No entanto, você ainda não tem certeza sobre qual parâmetro deve servir de base para a ordenação. Você tem três idéias:

- ordenar pelo começo
- ordenar pelo final
- ordenar pelo tempo de duração

Implemente o quicksort e utilize um algoritmo guloso que o utilize com base em um dos três parâmetros acima para resolver o problema.<sup>1</sup>

### Entrada

A entrada consiste de um número indefinido de conjuntos de dados, cada conjunto de dados consiste de uma linha com um número inteiro  $n_i$  ( $n_i \leq 100$ ) seguido de  $n_i$  linhas com dois inteiros distintos separados por espaço que denotam o intervalo. Cada inteiro é um número entre 0 e 23 que denota um horário. Os dados terminam com uma linha contendo apenas o número -1.

### Saída

A saída consiste de uma linha para cada conjunto de dados na entrada. Cada linha deve conter um único inteiro que representa o número máximo de intervalos disjuntos.

---

<sup>1</sup>Apenas um dos três métodos de ordenação explicitados resolve o problema para todos os casos.

## Exemplo de entrada-saída

entrada	saída
1	1
3 5	1
3	3
6 9	3
7 10	
8 11	
3	
1 2	
12 17	
2 4	
6	
3 6	
5 11	
4 5	
10 13	
7 10	
8 9	
-1	