

3.15. The police department in the city of Computopia has made all streets one-way. The mayor contends that there is still a way to drive legally from any intersection in the city to any other intersection, but the opposition is not convinced. A computer program is needed to determine whether the mayor is right. However, the city elections are coming up soon, and there is just enough time to run a *linear-time* algorithm.

(a) Formulate this problem graph-theoretically, and explain why it can indeed be solved in linear time.

5. (2.0pt) Um servidor recebe requisições de  $n$  clientes no instante 0, uma de cada cliente. Sabemos que o tempo que o servidor leva desde o instante que ele começa a atender a requisição do cliente  $i$  até o instante em que ele termina é  $t_i$ . Além disso, sabemos que o servidor atende apenas uma requisição por vez e que quando uma requisição começa a ser atendida ela não pode mais ser interrompida.

a) O tempo de espera de um cliente é definido como o tempo que ele espera até que sua requisição comece a ser atendida. Proponha um algoritmo eficiente para determinar a ordem em que o servidor deve atender os clientes de modo a minimizar o tempo médio de espera dos clientes. Analise a complexidade do algoritmo proposto.

b) Prove que seu algoritmo encontra a melhor solução possível.

2.23. An array  $A[1 \dots n]$  is said to have a *majority element* if more than half of its entries are the same. Given an array, the task is to design an efficient algorithm to tell whether the array has a majority element, and, if so, to find that element. The elements of the array are not necessarily from some ordered domain like the integers, and so there can be no comparisons of the form “is  $A[i] > A[j]$ ?”. (Think of the array elements as GIF files, say.) However you *can* answer questions of the form: “is  $A[i] = A[j]$ ?” in constant time.

(a) Show how to solve this problem in  $O(n \log n)$  time. (*Hint:* Split the array  $A$  into two arrays  $A_1$  and  $A_2$  of half the size. Does knowing the majority elements of  $A_1$  and  $A_2$  help you figure out the majority element of  $A$ ? If so, you can use a divide-and-conquer approach.)

[Hint: Suppose  $A$  has a majority element  $x$ . How many copies of  $x$  need to be in  $A_1$  or  $A_2$ ? What does this say about the majority of  $A_1$  and  $A_2$ ?]