Problema do Oráculo

Matheus Machado dos Santos 102449

Julho, 2014

1 Descrição do problema

Conceder um oráculo para derivar o limite inferior de um algoritmo para encontrar o máximo de uma lista de n elementos.

2 Resolução

Teorema: Qualquer algoritmo para encontrar o maior elemento de uma lista de n elementos não ordenados, $n \geq 1$, faz pelo menos $\lceil n/2 \rceil - 1$ comparações.

Prova: Utilizando um oráculo que descreve o comportamento de um algoritmo por meio de um conjunto de 3-tuplas, mais um conjunto de regras associadas que mostram as tuplas possíveis que um algoritmo pode assumir a partir de uma dada tupla e uma única comparação.

Uma 3-tupla, representada por (a, b, c), onfr os elementos:

- a - Elementos que nunca foram comparados.
- b - Elementos que sempre foram maiores em todas as comparações.
- c - Elementos perderam pelo menos uma vez em uma comparação.

O algoritmo inicial no estado $(n, 0, 0)$ e termina no estado $(0, 1, n-1)$, onde todos os elementos foram comparados e apenas um venceu todas as comparações.

As regras que determinam o comportamento de um algoritmo são as seguintes:

1. $(a - 2, b + 1, c + 1)$, uma comparação entre dois elementos de a.

2. $(a-1, b, c+1)$, uma comparação de a com b.
3. $(a, b-1, c+1)$, uma comparação entre dois elementos de b. com b.

A solução pode ser encontrada em dois passos:

1. Comparar todos elementos de a, 2 a 2, executando $\lceil n/2 \rceil$ vezes a regra 1 e resultando em $(0, n/2, n/2)$.
2. Comparar todos elementos de b, 2 a 2, aplicando $n/2 - 1$ vezes a regra 2, resultando em $(0, 1, n-1)$.

Logo, para obter o estado $(0, 1, n-1)$ a partir do estado $(n, 0, 0)$ são necessárias $n/2 + n/2 - 1 = n - 1$ comparações.