Universidade Federal de Minas Gerais
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Projeto e Análise de Algoritmos
3º Trabalho Prático - Paradigmas de Programação

A Spy History

Alberto Hideki Ueda

Orientador: Berthier Ribeiro Neto

Belo Horizonte
14 de novembro de 2014

1 Descrição do problema

Dada um mensagem composta de *bits* '0' e '1' e possíveis erros de transmissão, representados pelo símbolo '-', determinar se existe ou não um comando de controle na mensagem. Um erro de transmissão pode ser tanto um '0' quanto um '1'. Um comando de controle ou é uma sequência "000"ou uma sequência "11111". A resposta deve ser uma das seguintes:

$$Resposta = \begin{cases} true, \text{ se a mensagem com certeza contém um comando de controle;} \\ \\ false, \text{ se a mensagem com certeza não contém um comando;} \\ \\ both, \text{ se a mensagem pode ou não conter um comando, dependendo do que os erros significarem.} \end{cases}$$

2 Modelagem

A modelagem do problema é simples. Dada uma sequência de caracteres do conjunto $\{0, 1, -\}$ com tamanho máximo de 100 caracteres, determinar se tal sequência contém a subsequência "000" ou "11111"; ou então se é possível substituir os caracteres '-' da sequência por *bits* '0' e '1' de forma que exista alguma destas subsequências.

Há o caso particular em que mesmo que uma mensagem possua um ou mais caracteres '-', todas as formas de substituição possíveis contém subsequências de controle. Por exemplo, na mensagem "1111-00" todas as possíveis mensagens originais ("1111100" e "1111000") contém um comando de controle. Neste e nos demais casos em que este cenário se aplica, naturalmente a resposta do algoritmo deve ser *true*.

Se há pelo menos uma possibilidade da mensagem não conter tais subsequências **E** pelo uma possibilidade dela conter, a resposta deve ser *both*.

Caso não haja nenhuma possibilidade da cadeia de caracteres conter um comando de controle, a resposta deve ser $false$.

A seguir serão descritos os três algoritmos propostos para lidar com o problema, cada um seguindo um paradigma de programação diferente.

3 Algoritmo de Força Bruta

3.1 Complexidade de Espaço

3.2 Complexidade de Tempo

3.3 Análise Experimental

4 Algoritmo Guloso

4.1 Complexidade de Espaço

4.2 Complexidade de Tempo

4.3 Análise Experimental

5 Algoritmo com Programação Dinâmica

5.1 Complexidade de Espaço

5.2 Complexidade de Tempo

5.3 Análise Experimental

6 Dificuldades Encontradas

Figura 1:

7 Conclusão

Referências