Universidade Federal de Minas Gerais
Departamento de Ciência da Computação
Algoritmos e Estruturas de Dados III
1º Semestre de 2013

Trabalho Prático 3: MASLAB - Cadeado

Esse trabalho prático tem como objetivo colocar em prática o paradigma de programação dinâmica na resolução do problema de abertura de um cadeado.

Problema

Você e dois amigos (ambos engenheiros) decidem testar seus conhecimentos e participar da famosa competição de robôs organizada pelo *Mobile Autonomous Systems Laboratory*. Tal competição é conhecida por ser *o* torneio mais intenso de robôs do MIT. Tudo que você sabe sobre essa competição é que tarefas nas áreas de *Computer vision* e *maze navigation* são bastante comuns e portanto não foi surpresa nenhuma quando receberam o primeiro desafio: Construir e programar um robô que consiga, dada uma senha, abrir o cadeado de um armário. Foi deixado claro que a pontuação das equipes será em função do tempo gasto para solucionar o desafio, ou seja, quanto mais rápido mais pontos serão ganhos! Pensando na importância do tempo, você propõem a seus amigos que eles cuidem da parte funcional do robô enquanto você se preocupa com a lógica envolvida. Apos uma curta reflexão, você se da conta de que o "gargalo"deste problema, em questão temporal, estaria na lentidão na qual o robô executa a rotação das partes do cadeado para encontrar cada letra no lugar correto. Portanto, seu problema agora, se resume em calcular o menor número de movimentos necessários para abrir o cadeado, fazendo assim sua parte para chegar na solução mais rápida possível.

Um cadeado de senha, possui um sistema de código para ser aberto ao invés de uma chave. O cadeado em questão contém uma sequência de rodas, cada uma possuindo 26 letras do alfabeto inglês (a...z), em ordem. Se você move uma roda para cima, a letra que ela mostra muda para a próxima letra do alfabeto (se a letra mostrada for 'z', então ela muda para 'a'). Se você move uma roda para baixo, ela muda para a letra anterior do alfabeto (se a letra mostrada for 'a', ela muda para 'z'). É interessante perceber, que é possível mover qualquer subsequência contígua de rodas para a mesma direção com apenas um movimento. Isso tem o mesmo efeito que mover todas as rodas da subsequência para aquela direção, mas executando apenas um movimento. Exemplo: se a sequência secreta for a string 'zzzzzzzzz', será necessário apenas um movimento, ou melhor, mudar da letra 'a' para a letra 'z'.

O cadeado abre quando a roda mostra uma determinada sequência de letras. Lembre-se, de que, inicialmente, todas as rodas mostram a letra 'a'.

Entrada e Saída

A entrada contém vários instâncias de teste. Um caso de teste é descrito em exatamente uma linha contendo uma string não-vazia com no máximo 1000 letras minúsculas. A string representa a sequência secreta de letras que abre o cadeado.

Para cada caso de teste, será impressa uma linha contendo um único inteiro: o menor número de movimentos que abre o cadeado.

A entrada será lida de um arquivo e o resultado do programa deve ser impresso em outro arquivo de saída. Ambos arquivos devem ser passados por parâmetro na chamada do programa:

./tp3 input.txt output.txt

O arquivo de entrada possui um inteiro N na primeira linha, onde N é o número de instâncias a serem simuladas. Em seguida, as N instâncias devem conter uma string de tamanho $1 \leq M \leq 1000$. Cada string representa a sequência secreta para abrir o cadeado. Para cada instância, deve ser impresso no arquivo de saída, o menor número de movimentos que abre o cadeado.

Exemplo

A seguir temos um exemplo de funcionamento do algoritmo:

Entrada:

```
5
abcxyz
abcdefghijklmnopqrstuvwxyz
aaaaaaaaa
zzzzzzzzz
zzzzbzzzz
```

Saída:

```
5
25
0
1
3
```

Entrega

- A data de entrega desse trabalho é **20 de Junho**.
- A penalização por atraso obedece à seguinte fórmula $2^{d-1}/0.32\%$, onde d são os dias úteis de atraso.
- Submeta apenas um arquivo chamado <numero_matricula>_<nome>.zip. Não utilize espaços no nome do arquivo. Ao invés disso utilize o caractere '_'.
- Não inclua arquivos compilados ou gerados por IDEs. **Apenas** os arquivos abaixo devem estar presentes no arquivo zip.
  - Makefile
  - Arquivos fonte (*.c e *.h)
  - Documentacao.pdf
- Não inclua **nenhuma pasta**. Coloque todos os arquivos na raiz do zip.
- Siga rigorosamente o formato do arquivo de saida descrito na especificação. Tome cuidado com whitespaces e formatação dos dados de saída
- NÃO SERÁ NECESSÁRIO ENTREGAR DOCUMENTAÇÃO IMPRESSA!
- Será adotada **média harmônica** entre as notas da **documentação e da execução**, o que implica que a nota final será 0 se uma das partes não for apresentada.

Documentação

A documentação não deve exceder 10 páginas e deve conter pelo menos os seguintes itens:

- Uma **introdução** do problema em questão.
- **Modelagem e solução proposta** para o problema. O algoritmo deve ser explicado de forma clara, possivelmente através de pseudo-código e esquemas ilustrativos.
- **Análise de complexidade** de tempo e espaço da solução implementada.
- **Experimentos** variando-se o tamanho da entrada e quaisquer outros parâmetros que afetem significavelmente a execução.
- Especificação da(s) **máquina(s) utilizada(s)** nos experimentos realizados.
- Uma breve **conclusão** do trabalho implementado.
- Espera-se gráficos que explorem o tamanho da entrada do problema versus tempo de execução.

Código

- O código deve ser obrigatoriamente escrito na **linguagem C**. Ele deve compilar e executar corretamente nas máquinas Linux dos laboratórios de graduação.
- O utilitário ***make*** deve ser utilizado para auxiliar a compilação, um arquivo *Makefile* deve portanto ser incluído no código submetido.
- As estruturas de dados devem ser **alocadas dinamicamente** e o código deve ser **modularizado** (divisão em múltiplos arquivos fonte e uso de arquivos cabeçalho .h)
- **Variáveis globais** devem ser evitadas.
- Parte da correção poderá ser feita de forma automatizada, portanto **siga rigorosamente os padrões de saída especificados**, caso contrário sua nota pode ser prejudicada.
- **Legibilidade e boas práticas** de programação serão avaliadas.