

MC458 Projeto e Análise de Algoritmos I

Primeiro semestre de 2016

Docente: Orlando Lee

Monitora PED: Amanda Resende

Descrição

O espaço-porto (terminal) de Mos Eisley no planeta Tatooine armazena containers enormes que serão depois carregados em várias espaçonaves que os levarão para vários cantos da galáxia. Os containers chegam por estradas e são empilhados no próprio terminal à medida que chegam.

Espaçonaves carregam um grande número de containers. O tempo para carregar (colocar os containers dentro de) uma nave depende em parte da localização dos containers. O tempo de carregamento aumenta quando os containers não estão no topo da pilha, e só podem ser carregados após os containers que estavam sobre eles serem carregados. O terminal precisa de um plano (estratégia) para empilhar os containers de modo a diminuir o tempo de carregamento. Tal plano deve permitir que cada nave seja carregada apenas acessando containers que estão nos topos das pilhas, e minimizar o número de pilhas usadas.

Para este problema, sabemos a ordem na qual as naves devem ser carregadas e a ordem em que os containers chegam. Cada nave é representada por uma letra maiúscula entre A e Z (ou seja, há 26 naves) e as naves devem ser carregadas seguindo a ordem alfabética. Cada container é rotulado com uma letra maiúscula representando a nave na qual ele deve ser carregado. Não há limites no número de containers que podem ser colocados em uma pilha.

A sequência de chegada dos containers é descrita por uma sequência de letras maiúsculas. Por exemplo, a linha ABAC significa que 4 containers chegam consecutivamente e devem ser carregados nas naves A, B e C, respectivamente. Quando todos os containers chegarem, as naves são carregadas em ordem estritamente crescente: primeiro A, depois B, depois C e assim por diante.

Entrada

A entrada consiste de uma única linha contendo de 1 a 1000 letras maiúsculas representando a ordem de chegada de um conjunto de containers.

Saída

O número mínimo de pilhas necessárias para empilhar todos os containers antes do carregamento das naves começar.

Exemplo 1

```
A      /* entrada */  
1      /* saída */
```

Exemplo 2

CBACBACBACBACBA

3

Exemplo 3

CCCCBBBBAAAA

1

Exemplo 4

ACMICPC

4

Especificações

O programa deve ser implementado em C ou C++. Neste projeto é permitido o uso de quaisquer estrutura de dados ou rotinas de uma biblioteca do C ou C++.

Avaliação

Haverá 15 testes abertos e 5 teste fechados. A nota do projeto é proporcional ao número de teste bem-sucedidos. Mais precisamente, se N é o número de testes bem-sucedidos do seu programa, então sua nota será $(\frac{N}{20}) \times 10 = \frac{N}{2}$.

Prazo de submissão

O programa pode ser submetido até as seguintes datas:

- até 11:59h de 21 de março - sem penalidade,
- até 11:59h de 22 de março - serão descontados -4 pontos da nota, ou seja, a nota será $(\frac{N}{2}) - 4$.

Observações

- O número máximo de submissões é 10.
- Para a avaliação será considerada apenas a última versão do programa submetido.
- Note que é sempre possível fazer o serviço com 26 pilhas. Melhor ainda, é possível empilhar tudo usando um número de pilhas igual ao número de letras distintas da entrada. Entretanto, às vezes, pode-se fazer melhor (veja o **Exemplo 4**).
- Para submissão no Susy utilize o número do seu RA em **Usuário** e a senha utilizada na DAC em **Senha**.