UNIVERSIDADE FEDERAL DE SÃO JOÃO DEL REI
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO
PROJETO E ANÁLISE DE ALGORITMOS

1º semestre de 2024
Professor: Leonardo Chaves Dutra da Rocha

Trabalho Prático 3
Data de Entrega: 13 de Agosto 2024. Trabalho Dupla

Esse trabalho prático tem por objetivo ampliar os conhecimentos adquiridos em sala sobre processamento de cadeias de caractéres. O aluno terá a oportunidade de implementar e comparar vários métodos de casamento de padrões.

A Busca Mágica pela Substring Perdida

Em um reino encantado onde a magia permeia todas as coisas, uma antiga profecia foi revelada. Dizia-se que uma poderosa substring, capaz de conceder imenso poder a quem a dominasse, estava perdida nas profundezas das palavras místicas. Os sábios do reino, temerosos dos perigos que essa substring poderia desencadear nas mãos erradas, convocaram os mais habilidosos buscadores para recuperá-la.

Você, um jovem aprendiz de magia, foi escolhido para embarcar nessa jornada. O Grande Mago, guardião dos segredos ocultos, lhe confiou uma tarefa crucial: encontrar a substring perdida dentro de uma série de palavras encantadas.

Você recebeu um pergaminho mágico contendo uma string misteriosa e a substring que deve ser buscada. Além disso, foram-lhe concedidas N pergaminhos adicionais, cada um contendo um intervalo de palavras a serem investigadas. Se a substring estiver presente em uma palavra dentro do intervalo indicado, você deve revelar sua localização.

Mas cuidado! As palavras encantadas podem se ocultar em lugares sombrios e intricados, e somente os mais astutos e habilidosos serão capazes de desvendar seus segredos.

Com sua varinha mágica em mãos e sua mente afiada, você parte em uma jornada pelo reino, enfrentando desafios e perigos, na busca pela substring perdida. Será você capaz de decifrar os mistérios das palavras encantadas e encontrar o tesouro oculto antes que caia nas mãos erradas? A aventura aguarda!

Para realizar o match das strings devem ser implementados os algoritmos de casamento aprendidos em sala de aula.

Input

A primeira linha de entrada contém uma string de tamanho n, a segunda linha contém uma string de tamanho m, a terceira linha contém um inteiro k: o número de queries.

Após isso, a entrada possui k linhas descrevendo os queries. Cada linha contém dois inteiros a, b: o intervalo começa em a, termina em b

Output
Para cada querie, imprima *sim* caso a substring esteja presente e *nao* caso contrário.

Exemplo
Entrada:

```
abbaab
ab
3
1 4
3 5
2 6
```

Saída:

```
sim
nao
sim
```

Na tela, o programa deve imprimir apenas os tempos de usuário e os tempos de sistema para comparação. Para avaliação do tempo, utilize as funções *getrusage* e *gettimeofday*.

Documentação

Deve ser clara e objetiva, descrevendo as soluções adotadas e justificando bem as escolhas realizadas. Devem possuir também uma análise de complexidade detalhada das soluções. Em termos de análise de resultados, avalie o desempenho e funcionamento de seus algoritmos para diversas configurações e avalie também o tempo de execução dos mesmos (compare-os). Lembre-se, o importante é você apresentar maturidade técnica em suas discussões.

Observações:

- O código fonte do trabalho deve ser submetido para compilação e execução em ambiente Linux, tendo como padrão os computadores dos laboratórios do DCOMP.
- Deve ser escrito na linguagem C (trabalhos implementados em outras linguagens como C++/Java/Python e outras não serão aceitos);
- As estruturas de dados devem ser alocadas dinamicamente e o código deve ser modularizado utilizando os arquivos .c .h.
- O utilitário Make deve ser utilizado para compilar o programa;
- A saída deve ser impressa no arquivo pedido seguindo estritamente o formato da especicação caso contrário o resultado será considerado errado;
- O arquivo executável deve ser chamado de **tp1** e deve receber como parâmetro apenas o nome do arquivo de entrada de dados. Não serão aceitos outros nomes de executáveis além dos mencionados.

- Faça seu código de forma legível

Avaliação

Deverão ser entregues:

- listagem das rotinas;
- documentação contendo:;
  - descrição das soluções e estruturas de dados utilizadas;
  - análise da complexidade das rotinas;
  - análise dos resultados obtidos.
  - a documentação não pode exceder 12 páginas.

Distribuição dos pontos:

- execução (E)
  execução correta: 80%
- estilo de programação
  código bem estruturado: 10%
  código legível: 10%
- documentação (D)
  comentários explicativos: 30%
  análise de complexidade: 30%
  análise de resultados: 40%

A nota final é calculada como a média harmônica entre execução (E) e documentação (D):

$$\frac{D * E}{\frac{D+E}{2}}$$