

GCT053- Estruturas de Dados

Professor: Dr Johnatan Alves de Oliveira
Instituto de Ciências, Tecnologia e Inovação (ICTIN)
Universidade Federal de Lavras

2024-1

TRABALHO PRÁTICO II e III

Pontos: 10

Esse trabalho prático contempla os dois últimos trabalhos da disciplina. O conteúdo cobrado é referente a: algoritmos de ordenação, vetores, matrizes, complexidade assintótica, recursividade, equações de recorrência e manipulação de arquivos.

Data de entrega: 09/05/2024 23:59

Detalhes do Trabalho

Instruções

1. Crie os arquivos de código fonte necessários para a solução da questão (arquivos .c e .h);
2. Compile na linha de comando usando `gcc *.c -o prog.exe -lm`
3. A sua aplicação deve usar os conceitos estudados na disciplina;
4. O que vale é o horário do Campus Virtual, e não do seu, ou do meu, relógio!!!
5. Todos os requisitos do trabalho devem ser cumpridos.
6. A entrega deverá acontecer na UFLA virtual. Todo o código-fonte deve ser enviado para ser avaliado. Não é permitido o uso de bibliotecas de terceiros;
7. Teste o seu código em outro computador ou site que execute código-fonte para garantir que não tenha algum erro de execução.
8. A entrega e condução do trabalho é individual, mas os alunos podem trocar informações entre si.
9. Crie uma pasta para a questão: PrimeiroNome-UltimoNome-Questao (exemplo: johnatan-oliveira-1).
10. Execute usando redirecionamento de entrada: `./prog.exe < entrada.txt`.
11. Apague os arquivos gerados na compilação e o arquivo de entrada (mantenha apenas os arquivos .c e .h).
12. Compacte a pasta criada no item 1.
13. A cada etapa, verifique se o resultado está conforme o esperado.

Questão 1 - 5 pontos

Parabéns! Você foi contratado por uma renomada empresa de entretenimento digital para desenvolver um novo e empolgante jogo de quebra-cabeça chamado “Palavra Mágica”. Neste jogo inovador, os jogadores são desafiados a usar sua criatividade e habilidades linguísticas para desvendar uma sequência de caracteres mágicos. A missão é descobrir todas as combinações possíveis dessas letras encantadas para liberar feitiços especiais e avançar no jogo. Cada combinação correta revela um novo feitiço, proporcionando uma experiência mágica e envolvente.

Contexto do Jogo

Os jogadores estão em uma terra encantada e, para avançar pelos diferentes níveis, eles devem descobrir todas as palavras mágicas possíveis de uma sequência dada. Cada nível fornece uma sequência diferente e, para cada sequência, todas as combinações de caracteres devem ser encontradas.

Entrada

A entrada começa com um inteiro na primeira linha, N , representando o número de níveis (casos de teste). Cada um dos próximos N casos é uma linha contendo uma sequência de caracteres mágicos sem espaços.

Saída

Para cada nível (caso de teste), o programa deve gerar e imprimir todas as possíveis permutações da sequência de caracteres mágicos, respeitando o seguinte formato:

Nível 1:

<palavra mágica 1>

<palavra mágica 2>

:

<palavra mágica X>

Nível 2:

<palavra mágica 1>

<palavra mágica 2>

:

<palavra mágica Y>

Exemplo de Entrada

```
2
abc
123
```

Exemplo de Saída

```
Nível 1:
abc
acb
bac
bca
cab
cba
```

```
Nível 2:
123
132
213
231
312
321
```

OBS: Excepcionalmente para este caso, a saída de seu programa não precisa ser exatamente igual à saída esperada, no que diz respeito à ordenação das sequências.

Questão 2 - 5 pontos

Você foi contratado por uma empresa de jogos de aventura para criar um novo desafio para seu popular jogo "Exploradores da Ilha". Neste jogo, os jogadores precisam encontrar o caminho até o tesouro escondido em uma ilha representada por uma matriz. A matriz contém diferentes tipos de terrenos e obstáculos, e os jogadores só podem se mover em quatro direções: para cima, para baixo, para a esquerda e para a direita. Sua tarefa é implementar um algoritmo que determine se existe um caminho entre o ponto de entrada e o ponto onde o tesouro está escondido.

Implemente um programa para determinar se existe pelo menos um caminho entre o ponto de entrada 'E' e o ponto onde o tesouro 'T' está escondido em uma matriz que representa a ilha. Os movimentos permitidos são: Direita, Esquerda, Acima e Abaixo.

Entrada

A entrada começa com um inteiro na primeira linha, L , representando o número de mapas a serem explorados. A seguir, cada mapa é definido pelo seguinte padrão: na primeira linha é definido N , que representa a dimensão da matriz ($N \times N$); nas N linhas seguintes é definida a matriz, composta pelos seguintes símbolos:

- 0: terreno livre
- 1: obstáculo, não pode ser acessado
- E: ponto de entrada
- T: ponto onde o tesouro está escondido

Saída

A saída apresenta as respostas para cada mapa, seguindo o padrão:

Mapa 1: <resposta para o mapa 1>
Mapa 2: <resposta para o mapa 2>
:
Mapa L: <resposta para o mapa L>

Exemplo de Entrada

```
2
8
1 1 1 1 1 0 0 0
1 0 0 0 1 0 0 0
E 1 1 1 0 0 0 0
1 0 0 1 1 1 1 1
1 0 0 0 0 0 0 1
1 1 1 0 1 1 0 1
1 1 1 0 0 0 T 1
0 0 0 0 0 0 0 1
7
1 1 1 1 1 0 0
1 0 E 0 0 0 0
1 1 1 1 1 0 0
1 0 0 0 0 1 0
1 0 1 1 1 0 0
T 0 0 0 1 1 0
1 1 1 1 1 1 0
```

Exemplo de Saída

Mapa 1: SIM

Mapa 2: NAO

Questão 3 - Ordenação - 5 pontos

Você foi contratado como desenvolvedor por uma empresa de tecnologia que organiza grandes eventos esportivos. Sua primeira tarefa é desenvolver um sistema que ajude a organizar a lista de participantes dos eventos esportivos em ordem alfabética. Com a aproximação do evento, é crucial que os dados dos participantes sejam processados rápida e eficientemente. O algoritmo Merge Sort foi escolhido devido à sua eficiência e estabilidade para ordenar grandes conjuntos de dados. Para fins de análise e otimização do sistema, é necessário saber o número total de comparações realizadas durante a ordenação. Implemente um programa que utilize uma versão modificada do algoritmo Merge Sort para ordenar uma lista de nomes de participantes em ordem alfabética e contar o número de comparações realizadas durante o processo de ordenação. Logo, seu objetivo é Implementar uma versão modificada do algoritmo Merge Sort que, além de ordenar uma lista de nomes de participantes em ordem alfabética, também conte o número total de comparações realizadas durante a ordenação.

Entrada

A entrada começa com um inteiro na primeira linha, N, representando o número de participantes. A seguir, são definidos os N nomes dos participantes, um nome por linha.

Saida

A saída apresenta a lista de participantes ordenada em ordem alfabética e o número total de comparações realizadas durante a ordenação.

Exemplo de Entrada

```
5
Carlos
Ana
Beatriz
Eduardo
Daniel
```

Exemplo de Saída

Ana
Beatriz
Carlos
Daniel
Eduardo
Comparações: 7

Questão 4 - Complexidade - 5 pontos

Essa seção é específica sobre análise assintótica e equação de recorrência.

Equação de Recorrência

- a) Formule a equação de recorrência que descreve o tempo de execução dos algoritmos utilizados nas questões 1 e 3.
- b) Resolva a equação de recorrência para obter a complexidade temporal dos algoritmos das questões 1 e 3. Resolva a equação de recorrência via **Método da Substituição** e **Teorema Mestre**.