Problemas de Aplicação de Matemática para a Computação

Fabrício Galende Marques de Carvalho

OBSERVAÇÃO: Na solução dos exercícios e problemas seguintes, prepare uma pequena apresentação explorando os conceitos envolvidos no problema/exercício, justificando e explicando em detalhes sua resposta. Além disso, caso o problema requeira o desenvolvimento de um programa computacional e não tenha sido mencionada nenhuma linguagem de programação, utilize a linguagem de programação TypeScript e forneça como resposta o código-fonte, o procedimento de execução e o arquivo referente às dependências (e.g., package.json, etc). Caso seja necessária a preparação de uma interface para a visualização de dados, utilize o navegador e desenvolva o front end em JavaScript, HTML e CSS, consumindo os dados do back end utilizando express e banco de dados PostgreSQL.
Em todos os problemas dessa listagem, separe o arquivo contendo a definição do modelo do arquivo que faz uso (interface com o usuário). Modelos devem estar localizados em model.ts e interfaces (prompts ou interfaces gráficas) devem estar em view.ts.

1 Introdução a Matemática para a Computação

1.1. Um dos teoremas mais famosos que existe é o teorema de Pitágoras. Segundo esse teorema a soma dos quadrados dos catetos de um triângulo retângulo é igual a soma do quadrado da hipotenusa. Considerando esse teorema proceda com o seguinte:

(a) Efetue a prova matemática desse teorema utilizando similaridade de triângulos.

(b) Efetue a prova desse teorema utilizando a prova por construção gráfica (um quadrado menor de lado c dentro de um quadrado maior de lado (a+b).

(c) Efetue a prova matemática da recíproca do teorema, ou seja, se um triângulo possui um dos lados com medida tal que seu quadrado é igual a soma dos quadrados dos outros dois, então trata-se de um triângulo retângulo (use congruência).

(d) Ilustre um problema prático onde uma forma geométrica não elementar (sem ser um quadrado, um triângulo ou uma circunferência) pode ter sua área calculada através da composição de triângulos retângulos.

(e) Desenvolva um programa que recebe como entradas as potenciais medidas dos lados de um triângulo e checa se:

i. os lados podem representar um triângulo.

ii. os lados correspondem a um triângulo retângulo.

2 Definições Matemáticas

2.1. Para esse problema você deverá proceder com a utilização de definição por enumeração e também usando gênero e diferença. Para todas as construções de classe, não esqueça de utilizar apropriadamente os métodos de construção dos objetos.

(a) Efetue a definição dos meses do ano utilizando uma enumeração denominada Meses, usando cadeia de caracteres.

(b) Fazendo o uso da enumeração anterior, defina, utilizando gênero e diferença, uma classe do tipo Pessoa que deve possuir como atributos: dia de nascimento (número), mês de nascimento (valor do tipo enumeração), ano de nascimento (número) e nome (cadeia de caracteres). Essa classe deve possuir um método chamado apresentar-se, que imprime os atributos da pessoa em uma frase estruturada (ex. "Meu nome é fulano, nasci no dia tal do ano tal").

(c) Especialize essa classe para outra chamada Funcionário que, além de conter tudo que uma pessoa contém, também possui um número de matrícula, o nome da empresa onde trabalha (cadeia de caracteres), uma data de admissão (análoga à data de nascimento da classe Pessoa) e o nome do cargo que ocupa (ex. Desenvolvedor, engenheiro, etc). Essa classe deve possuir um método denominado descrever vínculo, que imprime o vínculo de acordo com os atributos da classe especializada.

(d) Especialize novamente essa classe para outra denominada Gestor, que além de conter tudo que um funcionário já contém, deverá conter nível de acesso. Nesse caso, o nível de acesso deverá ser representado por uma enumeração com valores de 1 a 3 que correspondem a "básico", "administrativo"e "irrestrito". O método que essa classe deve conter deve se chamar mostrar credenciais. Nesse caso, o objeto deve apresentar-se, citar seu vínculo e citar seu nível de credencial, considerando todos os métodos das classes-mãe.

(e) Crie um programa que possua um menu de cadastro que questiona sobre os dados necessários para um cadastro e prossegue preenchendo os atributos de acordo com o tipo requerido (pessoa, funcionário ou gestor). Ao final, o objeto do tipo apropriado é instanciado e os métodos são chamados a partir da instância da classe mais especializada (ex. Se o cadastro for de um gestor, esse deve se apresentar, descrever o vínculo e mostrar suas credenciais).

2.2. Desenvolva um algoritmo e o código-fonte que resolva, de modo recursivo, as seguintes operações matemáticas:

(a) O cálculo do fatorial.

(b) O cálculo de uma potência natural para um número real > 0.

(c) O cálculo do n-ésimo termo da série de Fibonacci.

(d) O cálculo do n-ésimo termo onde cada um dos termos, a partir do terceiro, depende da soma dos três termos anteriores.

Para cada um dos problemas citados ilustre a árvore de chamada, ou pilha de chamadas recursivas, e faça uma estimativa do uso de memória (análise de complexidade espacial aproximada). Ilustre, também, o caso onde ocorre o estouro de pilha. Compare os quatro algoritmos e faça observações sobre desempenho (tempo). Na sua resposta, não faça uso de cache e utilize orientação a objetos na modelagem e solução.

2.3. Desenvolva um pequeno sistema que ilustre a solução do problema das torres de Hanoi. O sistema recebe como entrada o número de discos, o pino de origem, o pino auxiliar e o pino de destino. O algoritmo deve ser desenvolvido de modo recursivo e deve ser ilustrado o passo a passo da solução. Desenvolva, também, uma pequena interface web que mostre cada uma das rodadas de execução do programa, ou seja, desde a configuração inicial até a configuração final, passando por cada um dos movimentos executados. Ilustre a operação para o caso de três pinos e as seguintes quantidades de discos: 1, 2, 3, 4, 5, 6 e 7.

3 Teoria de Conjuntos

3.1. Operar com números de magnitudes muito grandes ou muito pequenas pode gerar problemas tais como overflow, underflow e o chamado cancelamento catastrófico. Para evitar que tal problema ocorra, com frequência operações básicas de cálculo tais como soma, subtrações, multiplicações e divisões são emuladas em software ao invés de operadas diretamente no hardware, que é sujeito a limitações de representação. Desenvolva uma super calculadora, que seja capaz de efetuar somas e diferenças com elevadíssima precisão. Na sua implementação, utilize um array para armazenar tanto os números antes como os números após a vírgula (i.e., armazene cada dígito do número em uma posição do array e efetue a operação em cada um dos dígitos de modo a não perder precisão). Sua calculadora deve possuir a capacidade de operar com até 50 dígitos antes do ponto e até 50 dígitos após o ponto decimal. Desenvolva a interface de entrada em HTML que recebe dois números de ponto flutuante e retorna ou a soma ou a diferença entre esses números, de acordo com a escolha do usuário. O processamento deve ser efetuado em um back end todo desenvolvido em TypeScript. Ilustre a operação para casos tais como

(a) (1.1111111111111111111 + 1.1111111111111111111)

(b) (1000000000000.5 - 0.0000000000005)

3.2. A cifra de César é uma técnica rudimentar de criptografia baseada em operações básicas de soma, subtração e resto de divisão inteira. Desenvolva um pequeno sistema, composto por um front end em HTML e CSS e um back end, em TypeScript que receba como entrada um texto aberto em português e que retorne o texto cifrado utilizando a cifra de César. A chave deve ser informada pelo usuário e os valores correspondentes a cada um dos caracteres deve ser informada em um arquivo de configuração para o back end da aplicação (e.g., 'a':1, 'b':2, ..., 'A': 27, ...). Descreva os detalhes do algoritmo e como a cardinalidade do número de caracteres permitido afeta o número de possibilidades para a chave criptográfica.

3.3. Baseando-se na explicação fornecida pelo professor e no código-fonte disponibilizado no repositório da disciplina, desenvolva uma pequena aplicação que ilustre diferentes valores de Julia e Mandelbrot *Sets*. Seu programa deve receber como entrada os valores para as componentes reais e imaginárias, o passo a ser considerado para considerar os pontos no plano e o tipo de conjunto a ser gerado. A saída deve ser o fractal correspondente contendo pelo menos uma escala de 6 cores, de acordo com a magnitude e a iteração onde houve a divergência do conjunto.

3.4. A similaridade de Jaccard é uma técnica comumente utilizada em processamento de linguagem natural (PLN) para avaliar a similaridade entre dois documentos. Por definição, a similaridade de Jaccard é definida como sendo:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{1}$$

Considerando que A e B são os conjuntos representativos das palavras presentes em cada um dos documentos, desconsiderando caracteres de pontuação, espaços em branco e maiúsculas, desenvolva um sistema, composto de front end em HTML e CSS e um back end em JavaScript, que receba como entrada dois documentos textuais e que retorne a similaridade de Jaccard. Dê um exemplo prático de utilização do seu sistema para a detecção de plágio e aponte as possíveis limitações dessa técnica.

3.5. Desenvolva uma pequena aplicação que funcione como uma máquina de busca para voos de companhias aéreas. Uma tabela, denominada COMPANHIA AEREA possui o nome da companhia e as cidades nas quais elas operam (e.g., latam — manaus, latam — salvador, gol—porto alegre, etc.). Outra tabela, denominada CIDADES, possui cidades e turnos nos quais os voos são operados (e.g., manaus — noite, manaus – tarde, são paulo – manhã, etc.). A entrada para sua aplicação deve ser uma string de busca (e.g., "voos latam", "voos latam e manhã", "voos latam manaus", "voos latam ou noite", "voos gol exceto noturnos"). Essa string deve ser processada e mapeada para consulta envolvendo uma ou duas tabelas, utilizando *joins*, *unions* e *excepts*. A resposta deve ser retornada ao usuário de acordo com a busca solicitada. Considere que certas palavras da busca do usuário podem ser utilizadas para definir a operação entre os resultados das consultas individuais (e.g., e, ou e exceto).

3.6. A manipulação de dados geoespaciais é fundamental para a tomada de decisão e para a construção de sistemas de análise de dados modernos. Desenvolva um sistema que receba como entrada

o número de pessoas localizadas em uma determinada região (e.g., polígono representativo da área de construção da FATEC Jacareí), as coordenadas do polígno representativo da região e que retorne a densidade de pessoas localizadas naquela região (pessoas/m^2). Explique como esse sistema pode ser utilizado para calcular a densidade populacional tal como feito pelo IBGE no CENSO Demográfico.

3.7. Desenvolva um sistema composto por um mapa (tecnologia Leaflet) onde o usuário selecione uma origem e um destino e, utilizando as operações de distância efeuadas em banco de dados, o sistema informe a distância entre a origem e o destino e o tempo médio para se chegar ao destino considerando um trajeto em linha reta. Efetue a comparação com a informação fornecida por um aplicativo tal como o Google maps. Utilize como casos de teste trajetos que possam, de fato, ser percorridos em linha reta.

3.8. Um modelo de linguagem rudimentar é o chamado Naive Bayes. Esse modelo estima uma palavra de acordo com a palavra precedente. As estimativas de probabilidades dependem das contagens das ocorrências dos pares de palavras de acordo com os conjuntos de frases utilizadas no treinamento. Desenvolva um modelo simples de geração de frase baseado no Naive Bayes. No seu modelo, considere que todas as palavras são escritas em letras minúsculas. Seu modelo deve receber como entradas as frases de treinamento. Como saída, o modelo deve possuir um método $gerar\_frase(n : inteiro)$ que gera uma frase contendo n palavras de acordo com o modelo que foi treinado.