

Algoritmos Probabilísticos

O que é um algoritmo probabilístico?

Algoritmos probabilísticos são algoritmos que tomam decisões aleatórias durante a execução:

-Sobre uma mesma instância, um algoritmo probabilístico pode fornecer resultados diferentes em tempos diferentes.

Por que utilizar um algoritmo probabilístico?

Pois, por vezes, é preferível tomar uma decisão rápida aleatoriamente do que perder tempo calculando qual a melhor alternativa.

Cite algumas propriedades dos algoritmos probabilísticos:

-Podem dar respostas diferentes, vindas da execução da mesma instância de um problema.

-O tempo de respostas para cada execução pode variar gradativamente.

-Se há mais de uma resposta correta, pode-se encontrar outras respostas executando o algoritmo outras vezes.

-Podem gerar resultados errados

-Embora gerem resultados errados, não significa que são incertos, tendo a possibilidade de controlar a probabilidade de erro.]

-Como conseguimos controlar a probabilidade de erro, caso temos uma probabilidade menor que a de um algoritmo determinístico, e se o tempo de execução do Algoritmo probabilístico for menor que o do determinístico, o algoritmo probabilístico é mais confiável.

-Um algoritmo probabilístico é menos sensível a piores casos.

-Para a sua implementação é necessário o uso de geradores pseudorandomicos(algoritmos determinísticos que geram sequências de números a partir de uma semente).

Cite as principais categorias dos algoritmos probabilísticos:

-Algoritmos Numéricos

-Algoritmo de Monte Carlo

-Algoritmo de Las Vegas

O que são Algoritmos Numéricos?

Um algoritmo numérico é tal que fornece uma resposta associada a um intervalo de confiança. São aqueles que fornecem um resultado aproximado, e sua precisão aumenta conforme aumenta o número de execuções do algoritmo.

Cite exemplos de aplicação de algoritmos Numéricos:

Agulha de Buffon.

Integrais.

Contagem probabilística.

O que são Algoritmos de Monte Carlo?

Um algoritmo de monte carlo fornece a resposta correta exata com alta probabilidade, mas as vezes fornece uma resposta errada. Não se sabe se a resposta é certa ou errada.

Cite exemplos de aplicação de algoritmos de Monte Carlo:

Verificação de multiplicação de matrizes.

Teste de Primalidade.

O que são Algoritmos de Las Vegas?

São algoritmos parecidos com a técnica utilizada pelo monte carlo, porém verificam suas respostas, informando o erro no caso de falha, e caso falhar podemos simplesmente rodar novamente o algoritmo.

Comparação:

Em que ano o Brasil foi descoberto?

– Algoritmo Numérico

- Entre 1490 e 1510
- Entre 1493 e 1509
- Entre 1495 e 1506
- Entre 1499 e 1502

– Algoritmo Monte Carlo

- 1500, 1500, 3598, 1500, 1500, 1498, 1500, 1500

– Algoritmo Las Vegas

- 1500, ERRO, 1500, 1500, ERRO, 1500, 1500

Defina tempo médio e tempo esperado.

O tempo médio de execução é dado pela média de tempo consumido pelo algoritmo quando todas as instâncias de entrada são igualmente prováveis.

O tempo esperado de execução é dado pelo tempo médio que o algoritmo consumiria para resolver a mesma instância diversas vezes, embora haja a chance de ele levar um tempo equivalente ao pior caso para a sua execução.

Uma aplicação de algoritmos probabilísticos?

Criptografia, devido a independência a instancias especificas. O que impede que um atacante construa uma independência especifica que permita inferir algo sobre o algoritmo.

Classes de Complexidade

O que são classes de complexidade?

É um conjunto de problemas relacionados aos recursos computacionais baseados em complexidade.

O que é a classe P? Cite exemplos de problemas contidos na classe P.

Consiste nos problemas que podem ser resolvidos por uma máquina de turing determinística em tempo polinomial, mais especificamente em $O(n^k)$

- Caminhos Simples mínimos.
- Passeio de Euler
- Satisfazibilidade de 2-CNF.

O que é a classe NP?

Consiste nos problemas que são decidíveis que podem ser resolvidos por uma máquina de turing não-determinística em tempo polinomial. Se tivéssemos algum tipo de certificado de uma solução poderíamos verificar se o certificado é correto em tempo polinomial para o tamanho da entrada para o problema.

O que é a classe NP-Completo? Cite exemplos de problemas contidos na classe.

Consiste na classe de problemas que estão em NP e são tão difíceis quanto qualquer problema em NP. Estão confinados ao reino dos problemas de decisão, onde a resposta obtida é apenas “sim” ou “não”.

- Caminhos simples de comprimento máximo.
- Ciclo hamiltoniano.
- Satisfazibilidade de 3-CNF.
- Caxeiro viajante.
- Cobertura de vértices.
- Soma dos subconjuntos

Qual a vantagem de definir um problema como NP-Completo?

Se puder determinar que um problema é NP-Completo, estará dando uma boa evidência de sua intratabilidade.

Como provar que um problema é NP-Completo?

Através de 3 conceitos fundamentais que são: Problemas de decisão, Problemas de otimização e Reduções;

O que são problemas de decisão?

São problemas que a respostas é simplesmente “sim” ou “não”.

O que são problemas de otimização?

São problemas que cada solução possível, tem um valor associado e para os quais desejamos encontrar uma solução viável com o melhor valor.

O que são reduções? Como elas funcionam?

É a transformação de um problema em outro problema. Dada uma instância de α do problema A, use um algoritmo de redução de tempo polinomial para transformá-la em uma instância de β do problema B. Execute o algoritmo de decisão de tempo polinomial para B para a instância β . Use a resposta de β como a resposta de α .

Podemos utilizar a técnica de redução para provar que um problema desconhecido é NP-Completo, utilizando um problema NP-Completo conhecido.

O que é um ciclo Hamiltoniano?

Um ciclo hamiltoniano de um grafo não dirigido $G=(V, E)$ é um ciclo simples que contém cada vértice em V .

O que é um algoritmo de verificação?

É um algoritmo de dois argumentos A , onde um argumento é uma cadeia de entrada comum x e o outro é uma cadeia binária y denominada certificado.

Como contornar a NP-Completeness?

Temos no mínimo três formas de contornar a NP-Completeness. O primeiro é que, se as entradas reais são pequenas, um algoritmo com tempo de execução exponencial pode ser perfeitamente satisfatório. Também podemos contornar se isolarmos casos especiais importantes que podemos resolver em tempo polinomial. O terceiro é que poderíamos encontrar abordagens para determinar soluções quase ótimas, ou seja um algoritmo de aproximação.

Algoritmos Aproximativos

O que é um algoritmo aproximativo? Onde são utilizados?

É um algoritmo que retorna soluções quase ótimas. São utilizados para problemas de minimização e maximização.

O que é uma razão de aproximação? Onde são utilizadas?

Dizemos que um algoritmo para um problema tem uma razão de aproximação $p(n)$ se, para qualquer entrada de tamanho n , o custo C da solução produzida pelo algoritmo está a menos de um fator $p(n)$ do custo C^* de uma solução ótima. São utilizados para problemas de minimização e maximização.

O que é desigualdade triangular? Para que serve no escopo de algoritmos aproximativos?

A desigualdade triangular afirma que o comprimento de um dos lados é sempre inferior à soma dos comprimentos dos outros dois lados.

A desigualdade triangular é utilizada na função custo para poder afirmar que o modo menos caro de ir de um lugar “ u ” a um lugar “ w ” é ir diretamente, sem nenhuma etapa intermediária. E assim eliminando uma parada intermediária não aumentaria o custo do caminho.

Programação Linear

Em programação linear o grande objetivo é conseguir criar uma estratégia que consiga otimizar ao máximo a função linear, por isso programação linear, de acordo com o conjunto de desigualdades do problema.

EX:

- Problema do Politico

Um politico tenta ganhar uma eleição diminuindo os custos de sua campanha, colocando el um gráfico as propostas para a população (x) e áreas de abrangência (y) através de dados referentes aos tipo de proposta combinado com a zona urbana é possível saber qual investimento terá maior retorno, ou seja, versão mais otimizada.

Formas padrão e relaxada

Padrão: faz a maximização de uma função linear sujeita a desigualdades lineares

Relaxada: Faz a maximização de uma função linear sujeita a igualdades lineares.

Resumindo: Em geral utilizamos a forma padrão para expressar programas lineares, mas é mais conveniente usar a forma relaxada quando descrevemos algoritmos simplex.

Quais as aplicações de programação linear?

- Cenário eleitoral é um exemplo tipico, como descrevemos no problema do politico
- Uma empresa aérea precisa programar a carga horária de tripulações de forma eficiente seguindo restrições de **limite de horas de voo**, tripulação trabalhando sempre no mesmo **modelo de aeronave**, programação linear ajuda em fazer a escala com o menor numero possível de tripulantes.
- Empresa petrolífera na análise de uma perfuração consegue ter resultados em questão de custo/beneficio.

Quais algoritmos são caracterizados como de programação liner

- Algoritmo de elipsoide (executado lentamente na pratica)
- metodos do ponto anterior (tempo polinomial, é uma classe de algoritmos)