

Universidade Federal de Roraima
Departamento de Ciência da Computação
Análise de Algoritmos
DISCIPLINA: Análise de Algoritmos – DCC606
2ª Lista
Prazo de Entrega: 26/06/2018

Erick dos Santos Batista

[QUESTÃO – 01] Especifique cada problema e calcule o M.C. (melhor caso), P.C. (pior caso), C.M. (caso médio) e a ordem de complexidade para algoritmos (os melhores existentes e versão recursiva e não recursiva) para problemas abaixo. Procure ainda, pelo L.I. (Limite Inferior) de tais problemas:

- (A) N-ésimo número da sequência de Fibonacci
- (B) Geração de todas as permutações de um número

[QUESTÃO – 02] Defina e dê exemplos:

- (A) Grafos.

Na literatura existem diferentes definições, mas em todas dizem uma coisa em comum que seria, que um grafo possui um conjunto V de vértices e um conjunto E de arestas onde nos grafos direcionados temos mapas $s, t : E \rightarrow V$, onde $s(e)$ é a *fonte* e $t(e)$ é o *alvo* da aresta direcionada e . E nos não direcionados temos uma função $w : E \rightarrow \mathcal{P}(V)$ que associa a cada aresta um subconjunto de dois ou de um elemento de V , interpretado como os pontos terminais da aresta.

Podemos usar como exemplo de grafos uma rede de computadores.

- (B) Grafo conexo, acíclico e direcionado.

Um **grafo** $G=(V, E)$ é **conexo** se existir um caminho entre qualquer par de vértices

Um grafo acíclico e direcionado e um grafico cujo mesmo nao possui ciclos, ou seja, para qualquer vértice v , não há nenhuma ligação dirigida começando e acabando em v . Um exemplo desse tipo de grafo é uma linha, e na computação A estrutura de directoria num típico sistema de ficheiros UNIX.

- (C) Adjacência x Vizinhança em grafos.

Em um grafo simples (a exemplo de G_1) dois vértices v e w são adjacentes (ou vizinhos) se há uma aresta $a=(v,w)$ em G . Está aresta é dita ser incidente a ambos, v e w .

Um exemplo para vizinhança é um mapa, onde os vértices são casas e as arestas são as ruas, considerando uma casa A as casas vizinhas serão as casas onde exista uma rua ligada diretamente a A .

- (D) Grafo planar.

É um grafo que pode ser desenhado em um plano sem que suas arestas se cruzem.

- (F) Grafo completo, clique e grafo bipartido.

Grafo completo é um grafo simples que em cada par de vértices são conectados por uma aresta. O clique é um conjunto de vértices V , onde para cada par de vértices de V , existe uma aresta que os conecta, ou seja, um clique é um subgrafo no qual cada vértice é conectado a cada outro vértice do sub grafo, ou seja, todos os vértices do sub grafo são adjacentes. E grafo bipartido é um tipo de grafo $G = (N, E)$ onde seus vértices podem separar-se em dois conjuntos disjuntos U e V .

(G) Grafos simples x multigrafo x digrafo.

Um grafo simples é um grafo que não contém laços e arestas múltiplas. Já um grafo multigrafo é um grafo onde os vértices possuem arestas múltiplas. E o dígrafo é um grafo direcionado e simples.

[QUESTÃO – 03] Defina e apresente exemplos de matriz de incidência, matriz de adjacência e lista de adjacência. Adicionalmente, descreva o impacto (vantagens e desvantagens) da utilização de matriz de adjacência e lista de adjacência.

Matriz de incidência

Matriz de incidência é um grafo representando computacionalmente através de uma matriz bidimensional, onde uma das dimensões são vértices e a outra dimensão são arestas.

Ex: Dado um grafo G com n vértices e m arestas, podemos representá-lo em uma matriz $n \times m$ M .

A definição precisa das entradas da matriz, onde varia de acordo com as propriedades do grafo, porém de forma geral tem as informações sobre como os vértices se relacionam com cada aresta.

Para representar um grafo sem pesos e não direcionado, as entradas da matriz M devem conter 1 se a aresta incide no vértice, 2 se for um laço e 0, se não incidir no vértice.

Matriz de adjacência

Matriz de adjacência é uma forma de se representar um grafo.

Ex: Dado um grafo G com n vértices, representamos-o em uma matriz $n \times n$ $A(G)=[A_{ij}]$ (ou simplesmente A). A definição precisa das entradas da matriz varia de acordo com as propriedades do grafo que se deseja representar, porém de forma geral o valor A_{ij} guarda informações sobre como os vértices V_i e V_j estão relacionados (isto é, informações sobre a adjacência de V_i e V_j).

Para representar um grafo não direcionado, simples e sem pesos nas arestas, basta que as entradas A_{ij} da matriz A contenham 1 se V_i e V_j são adjacentes e 0, caso contrário. Se as arestas do grafo tiverem pesos, A_{ij} pode conter, ao invés de 1 quando houver uma aresta entre V_i e V_j , o peso dessa mesma aresta.

Algumas das vantagens e desvantagens de se usar matriz de adjacência e lista de adjacência são:

Em uma matriz de adjacência a velocidade de leitura e escritura é constante.

Já uma lista de adjacência ocupa menos espaço na memória.

Se a quantidade de vértices for igual ao número de arestas, utilizar a matriz de adjacência é mais viável na maioria dos casos.

Mas se o número de arestas é inferior ao número de vértices na maioria dos casos e melhor utilizar lista de adjacência.

[QUESTÃO – 04] Comente sobre tabelas hash, apresentando a complexidade para as operações realizadas. Adicionalmente, implemente um tabela hash com encadeamento separado usando: lista encadeada e árvore vermelho e preto. Apresente um estudo empírico para obter custos de inserção na medida em que o número de chaves aumenta. Gere gráficos para mostrar o custo de inserção para tamanhos distintos de N (exemplo: de 10 a 10000). Descreva uma análise de comparação em relação ao tempo de execução.

Tabela hash é uma estrutura de endereçamento que é usada para armazenamento, onde o rápido acesso aos dados a partir de um espaço enorme de chaves se torna possível. O mapeamento das chaves é feito através de uma função hash, assim o tempo $O(1)$.

Onde temos grandes problemas que são as colisões. Uma colisão ocorre quando duas ou mais chaves tem o mesmo valor hash. Podemos solucionar esse problema se pode utilizar técnicas de encadeamento aberto e encadeamento fechado. No caso de encadeamento fechado, se implementa uma subestrutura em cada posição da tabela hash para salvar as chaves com o mesmo valor hash ali, essas estruturas podem ser listas encadeadas, árvores AVL e outros. Temos com pior caso do hash por encadeamento $O(n)$. E no melhor, $O(\lg n)$.

[QUESTÃO – 05] Defina, explicando as principais características e exemplifique:

(A) Enumeração explícita x implícita.

Enumeração explícita e implícita são métodos que fazem enumerações de soluções para um determinado problema, o explícita enumera todas as possíveis soluções, e a implícita faz uma enumeração apenas com os melhores problemas.

Um exemplo de explícita seria ter um baralho na mesa, onde o jogador sabe o valor de cada carta, não precisando virar nem uma carta para saber qual carta é, já na implícita, a pessoa teria que virar a carta.

(B) Programação Dinâmica.

Aqui usamos um método chamado de divisão e conquista, onde temos como objetivo minimizar o todo o problema em partes, aplicamos programação dinâmica quando um subproblema aparece diversas vezes ao longo do processo, pois dessa forma esse subproblema só é resolvido uma vez.

Um bom exemplo é a sequência de Fibonacci:

(C) Algoritmo Guloso.

Algoritmos gulosos tendem a usar uma técnica de projeto de algoritmos que resolve o problema com decisões localmente ótimas, com a esperança de encontrar um ótimo global. Uma vez que ele escolhe o próximo passo, ele não volta atrás.

(D) Backtracking.

Esse algoritmo é um tipo de refinamento da busca gulosa, onde múltiplas soluções podem ser eliminadas sem serem explicitamente examinadas.

[QUESTÃO – 06] Implemente uma solução para multiplicação de matrizes utilizando programação dinâmica, visando determinar uma ordem em que as matrizes sejam multiplicadas, de modo a minimizar o número de multiplicações envolvidas.

[QUESTÃO – 07] Implemente uma solução para encontrar o caminho mínimo usando a técnica backtracking. Lembrando que o caminho mínimo consiste na: minimização do custo de travessia de um grafo entre dois nós (ou vértices), custo este dado pela soma dos pesos de cada aresta percorrida

[QUESTÃO – 08] Defina e exemplifique:

(A) Problema SAT x Teoria da NP-Completeness.

(B) Classes P, NP, NP-Difícil e NP-Completo.

[QUESTÃO – 10] Descreva a redução (prove a NP-Completeness) do problema do SAT ao Clique. Apresente o pseudo-código do algoritmo NP e mostre graficamente as instâncias e soluções, no processo de redução.