

Universidade Federal de Roraima
Departamento de Ciência da Computação
Análise de Algoritmos
DISCIPLINA: Análise de Algoritmos – DCC606
2ª Lista

Questão 1

Fibonacci: Função recursiva

Complexidade: $O(\Phi^n)$.

Melhor caso e pior caso: 2^n

Fibonacci: função iterativa.

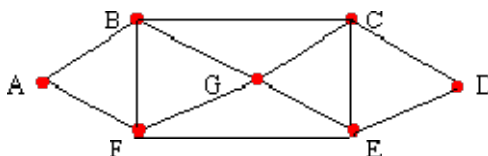
Complexidade: $O(n)$.

Melhor caso e pior caso: n .

Questão 2

Questão 3

(A) Grafos: Para Feofiloff, Kohayakawa, Wakabayashi, 2011: Um grafo é um par (V, A) em que V é um conjunto arbitrário e A é um subconjunto de $V \times V$. Os elementos de V são chamados vértices e os de A são chamados arestas. Neste texto, vamos nos restringir a grafos em que o conjunto de vértices é finito. Basicamente podemos definir como um conjunto de vértices (V) que são interligados por arestas (A) .



(B) Grafo conexo, acíclico e direcionado: Dizemos que um grafo é conexo se, para qualquer par $\{v, w\}$ de seus vértices, existe um caminho com extremos v e w .

Dizemos que um grafo é acíclico, quando para qualquer par $\{v, w\}$ de seus vértices, não existe um caminho que liga v e w sem que ao menos uma vez uma aresta não seja percorrida mais de uma vez.

Grafo direcionado ocorre quando cada aresta possui apenas um sentido de um vértice para outro, ou seja, (A) a aresta possui direção de $\{v \rightarrow w\}$ e o caso contrário não se aplica.

(C) Adjacência x Vizinhos em grafo: Um grafo $G=(V,E)$, onde V são os vértices e E suas arestas, os vértices ligados por uma aresta são considerados adjacentes ou vizinhos.

(D) Grafo planar: é um grafo que pode ser disposto onde as arestas não se cruzem.

(E) Grafo completo, clique e grafo bipartido: um grafo é completo quando existe uma aresta ligando qualquer par de vértice.

Um clique em um grafo não direcionado $G = (V, E)$ é um subconjunto de vértices $C \subseteq V$, tal que para cada dois vértices em C , existe uma aresta os conectando. Isso se equivale a dizer que um subgrafo induzido de C é completo (em alguns casos, o termo clique também pode ser referência ao subgrafo). Supondo que vários bairros formem um distrito, e que nossos bairros são os vértices e que as estradas que ligam esses bairros são as arestas, supondo que tenhamos uma cidade que possua vários distritos, o clique é o que vai dividir esses dois distritos em um novo distrito.

Um grafo é bipartido se for possível dividir tal grafo em dois grupos, sendo que cada vértice de um grupo só possui arestas para os vértices do outro grupo.

(F) Grafo simples x multigrafo x dígrafo: Grafo simples possui apenas uma aresta para cada par de vértices, já Multigrafo pode conter mais de uma aresta para o mesmo par de vértices, e o Dígrafo pode mais de uma aresta para cada par de vértices e essas arestas possuem direção.

Questão 4

Matriz de incidência: representa um grafo, onde as linhas e colunas representam os vértices e as arestas são os valores que irão preencher a posição A_{jk} , da seguinte forma:

“0” se não há incidência;

“+1” se o vértice k sai do vértice j ;

“-1” se o vértice k entra no vértice j .

Matriz de adjacência: é uma representação de um grafo de n vértices, onde cada elemento A_{ij} é determinado da seguinte maneira: “1” se uma aresta ligando o A_i e A_j e “0” se não há aresta.

Lista de adjacência: é uma representação de um grafo de n vértices, em que cada vértice possui uma lista e os elementos da lista são os seus vértices adjacente.

Comparação: Quando a quantidade de vértices é igual ao número de arestas utilizar matriz de adjacência se torna uma opção mais viável, quando o número de arestas é bem inferior ao número de vértices, opta-se por utilizar a lista de adjacência, porém cada caso tem que ser analisado.

Questão 6

(A) Enumeração explícita x implícita:

O método de enumeração explícita, faz uma enumeração de todas as possíveis soluções, já em enumeração implícita faz uma enumeração “inteligente”, com as melhores soluções para o problema.

(B) Programação Dinâmica: É a técnica de programação onde se vale dos resultados já obtidos e que são armazenados em uma tabela, para que no futuro se o mesmo caso ocorrer, não precisa refazer os cálculos, apenas acessar o valor armazenado, impactando no tempo de execução.

(C) Algoritmo Guloso: o algoritmo guloso não possui backtracking, não se vale de analisar as possíveis soluções como um todo, apenas o próximo passo a seguir,

(D) Backtracking: se vale de analisar todas as possíveis soluções, podendo retornar para outra solução caso aquela que esteja seguindo não seja a melhor, ele analisa o problema e a solução como um todo.