Perguntas teste 2

De que depende o desempenho do algoritmo de pesquisa binária baseado numa Árvore de Pesquisa Binária (BST)? *Justifique claramente*.

-O desempenho do algoritmo de pesquisa binária baseado numa Árvore BST depende da profundidade da árvore.

Diga o que entende por BST aleatória (*random BST*)? *Justifique claramente*.

- A construção de uma BST aleatória não necessita que os elementos sejam introduzidos de forma aleatória para que a topologia da árvore resultante seja equivalente. Para que isto seja possível é necessário inserir um elemento na raiz da BST. Desta forma, as random BST's vão reduzir fortemente a probabilidade de ocorrência do caso pior e garantir a eficácia no desempenho.

Diga que metodologias de resolução de colisões conhece (para tabelas de *hashing*) e quais as vantagens e desvantagens de cada uma delas. *Justifique claramente*.

- Existem 2 metodologias para a resolução de colisões: o encadeamento e o endereçamento aberto.
O encadeamento contém elementos que colidem em listas ligadas (slot). O processo de pesquisa é a soma da determinação do valor de hash e da pesquisa sequencial na lista ligada corresponde. Para guardar os elementos que colidem usam-se listas ordenadas e BST's Auto-Equilibradas.
No endereçamento aberto, para cada elemento que colide é procurada uma nova posição na tabela. A pesquisa é a soma da determinação do valor de hash e da pesquisa da nova posição do elemento na tabela.
Para que a resolução de colisões funcione é necessário que não haja qualquer colisão. Ou seja, a eficácia deste método depende do número de colisões, que depende da função de hash e do tamanho da tabela, comparativamente ao tamanho do Universo. Para haver uma boa função de hash é necessário que cada chave tenha a mesma probabilidade de corresponder a cada elemento da tabela, se isto não ocorrer as 2 metodologias não serão favoráveis.

Diga se o método de pesquisa baseada na profundidade DFS (*Depth-First Search*) para grafos serve para determinar o caminho mais curto entre dois vértices de um grafo. *Justifique claramente*.

-O método mais apropriado para determinar o caminho mais curto entre 2 vértices é a pesquisa baseada na amplitude BFS, pois deste método vai resultar numa árvore de pesquisa que pode ser utilizada na determinação de caminhos mais curtos.

A partir do vértice original vai verificar todos os caminhos possíveis e vai visitar todos os vértices ainda não visitados, até chegar ao vértice pretendido. O primeiro que chegar ao vértice que se pretende é o caminho mais curto.

Suponha que tem uma rede que pode ser modelada através de um grafo (não dirigido e não ponderado). Como faria para determinar a existência (ou não) de conectividade entre dois pontos dessa rede? *Justifique claramente*.

- O método mais adequado seria o BFS pois a árvore de pesquisa resultante deste método pode ser utilizado na determinação do caminho mais curto de um vértice 'v' para um vértice 'w'. Nestes caso, como queremos saber se 2 vértices estão ligados, o BFS faz a comparação entre estes 2 pontos enquanto o outro começa a partir de um ponto qualquer (vértice).

Qual o desempenho do algoritmo de pesquisa binária baseado na utilização de uma Árvore de Pesquisa Binária (BST), se esta for construída a partir de um grupo ordenado de elementos utilizando o método standard de inserção? Justifique claramente.

-A eficácia das BST como suporte de algoritmos de pesquisa depende do equilíbrio da referida BST. A pesquisa binária é uma pesquisa numa lista ordenada dividindo repetidamente o intervalo de pesquisa ao meio, desta forma este método tem um bom desempenho dependendo da profundidade da árvore, pois vai efetuar demasiadas vezes o mesmo processo até encontrar o valor ou até ficar com um intervalo vazio (quando a pesquisa não é bem sucedida).

Diga qual a vantagem de utilizar red-black trees se elas necessitam de mais informação na árvore (cor das ligações – e dos nós). Justifique claramente.

-Apesar de as red-black trees necessitarem de mais informação, estas vão ajudar pois como cada nó tem uma cor que representa a ligação que aponta para ele, desta forma, leva à diminuição da perda de eficácia devido à informação extra (cor).

Uma empresa regista as entradas dos seus funcionários através de um cartão do qual consta um código alfanumérico identificador único. Diga, justificando claramente, que cuidados deve ter se, para a identificação (pesquisa) de um funcionário através desse código, o algoritmo utilizar tabelas de endereçamento com funções de hash (hashing).

- As tabelas de endereçamento com funções de hash são favoráveis para universos com chaves pequenas. Para utilizar estas tabelas é necessário que nem todas as chaves colidam no mesmo lugar, nem utilizar a pesquisa sequencial, para que não ocorra o caso pior. Este método depende do número de colisões que dependem da função de hash e do tamanho da tabela, comparativamente ao tamanho do Universo. Para que haja uma boa função de hash é necessário que cada chave tenha a mesma probabilidade de corresponder a cada elemento.

Num algoritmo genético, o operador de mutação desempenha um papel fundamental. Explique a função deste operador, elaborando um esquema gráfico de modo a ilustrar o seu funcionamento.

-O operador mutação modifica aleatoriamente uma solução. Simula a ocorrência de erros que ocorrem com uma pequena probabilidade durante a duplicação de cromossomas. Os movimentos no espaço de pesquisa, a possível recuperação de informação perdida e a probabilidade não-nula de visitar todos os pontos do espaço de pesquisa são resultados práticos da mutação.

0|1|0|1|1|0|0|1|1|0|1|1|1|0|1|1

0|1|0|1|1|0|0|1|0|0|1|1|1|0|1|1

Diga se entende que o algoritmo de Prim pode ser considerado uma extensão do algoritmo BFS? Justifique claramente.

- Sim o algoritmo de Prim pode ser considerado uma extensão do algoritmo BFS pois utiliza uma fila de espera com prioridade.

Suponha que tem uma rede de comunicações que pode ser modelada através de um grafo (dirigido e ponderado). Como faria para determinar o melhor caminho entre dois pontos dessa rede? Que alterações teria de introduzir se o grafo não fosse dirigido? Justifique claramente.

- Utilizaria o algoritmo de Dijkstra que funciona visitando os vértices no grafo a partir de um ponto de partida. Examina respetivamente o vértice mais próximo da origem que ainda não foi examinado, adicionando os seus vértices ao conjunto de vértices a serem examinados (expande-se a partir do ponto de partida até que atinja o objetivo). Desta forma, garante que encontra o caminho mais curto de um ponto de partida para um outro vértice, desde que nenhuma ligação tenha custo negativo.

Caso o grafo não fosse dirigido, usaria o algoritmo de Prim pois este é recomendado para grafos densos como é o caso da rede de comunicação.

Quais os cuidados a ter ao criar e manter uma Árvore de Pesquisa Binária (BST)? *Justifique claramente*.

-Ao criar uma BST tem de se ter em atenção à criação dos nós. Todos os nós da sub-árvore esquerda têm valores menores ou iguais ao pai e todos os nós da sub-árvore da direita têm de ter valores maiores ou iguais ao pai. A eficácia das BSTs como suporte de algoritmos de pesquisa depende do equilíbrio da referida BST.

Ao utilizar tabelas de endereçamento com funções de *hash* (*hashing*) para efectuar pesquisas em conjuntos de dados, que cuidados deve ter para minorar o problema das colisões, se estas forem resolvidas com endereçamento aberto? *Justifique claramente*.

- O endereçamento aberto é utilizado sempre que se pode estimar à partida o número de elementos na tabela e este é folgado em relação à dimensão desta. Neste caso, o processo de pesquisa é a soma da determinação do valor de hash e a pesquisa da nova posição do elemento na tabela. Como para cada elemento que colide é procurado uma nova posição na tabela, temos que ter em atenção ao tamanho desta em relação ao número de elementos.

Em suma, o endereçamento aberto é uma boa opção mas faz um uso pouco eficaz da memória devido à memória que é reservada. Desta forma, quando são vários elementos, a melhor opção é através do encadeamento pois não tem memória ocupada quando se pretende inserir mais elementos.

Uma empresa emissora de cartões de crédito, precisa de desenvolver um software que identifique o cliente através do número do cartão. Diga, *justificando claramente*, se a utilização de tabelas de *hashing* neste algoritmo de pesquisa é adequada.

- Neste caso, a utilização de tabelas de hashing é adequada, pois as tabelas de endereçamento com funções de hash são favoráveis para universos com chaves pequenas. Para utilizar estas tabelas é necessário que nem todas as chaves colidam no mesmo lugar, nem utilizar a pesquisa sequencial, para que não ocorra o caso pior. Este método depende do número de colisões que dependem da função de hash e do tamanho da tabela, comparativamente ao tamanho do Universo. Para que haja uma boa função de hash é necessário que cada chave tenha a mesma probabilidade de corresponder a cada elemento.

Determinada empresa de fornecimento de internet pretende desenvolver um software para determinar se dois pontos da sua rede estão ligados. Supondo que pode modelar a rede através de um grafo simples (não dirigido e não ponderado) diga, *justificando claramente*, que método utilizaria para resolver este problema.

-O método mais adequado seria o BFS pois a árvore de pesquisa resultante deste método pode ser utilizado na determinação do caminho mais curto de um vértice 'v' para um vértice 'w'. Nestes caso, como queremos saber se 2 vértices estão ligados, o BFS faz a comparação entre estes 2 pontos enquanto o outro começa a partir de um ponto qualquer (vértice).

Diga em que situações deverá preferir o algoritmo de Kruskal em vez do de Prim para a determinação da Árvore Abrangente de Custo Mínimo (*minimum spanning tree - MST*) de um grafo ponderado. *Justifique claramente*.

-O algoritmo de Prim baseia-se na ideia de manter um corte do grafo, dividindo-o em vértices da MST e vértices não pertencentes à MST. Este algoritmo é adequado para grafos densos e pode ser considerado uma "extensão" do algoritmo BFS pois utiliza uma fila de espera com prioridades.

O algoritmo de Kruskal constrói a MST adicionando uma ligação de cada vez (ligação fonte) e constrói a árvore ligação a ligação, encontrando a ligação que une duas árvores numa floresta. Este algoritmo inicia-se assim com uma floresta de árvores (1 vértice) e com a ordenação das ligações por custo, desta forma é um algoritmo ótimo para grafos com pouca densidade.

Em suma, a preferência do algoritmo de Kruskal em vez do de Prim depende da densidade dos grafos, pois o primeiro só irá ser preferido se os grafos tiverem pouca densidade.

Diga se entende que o algoritmo de Dijkstra pode ser considerado uma extensão do algoritmo BFS? *Justifique claramente*.

-O algoritmo de Dijkstra funciona visitando vértices no grafo a partir de um ponto de partida. Examina respetivamente o vértice mais próximo da origem, ainda não examinado, adicionando os seus vértices ao conjunto de vértices a serem examinados (expande-se a partir do ponto de partida até que atinja o objetivo). Desta forma, garante que encontra o caminho mais curto de um ponto de partida para um outro vértice, desde que nenhuma ligação tenha custo negativo. Logo podemos afirmar que o algoritmo de Dijkstra pode ser considerado uma extensão do algoritmo BFS.

Ao tentar desenvolver um software para determinar o caminho mais curto entre dois pontos com recurso a grafos, surgiu um problema: determinada rua/estrada (que corresponde a uma ligação entre dois vértices) só pode ser percorrida num sentido e, desse modo, esse percurso é sempre feito em descida, podendo ser considerado um percurso de custo negativo. Que implicações pode ter este facto no nosso algoritmo? *Justifique claramente*.

-Ao ser considerado um percurso de custo negativo isto vai ter implicações relativamente à utilização do algoritmo de Dijkstra, por exemplo, pois este não vai conseguir encontrar o caminho mais curto de um vértice para outro.

A operação de *splay* corresponde a:

-Mover um nó para a raíz.

No caso de resolução de colisões em processos de tabelas de hashing, e para o caso em que o número de elementos da tabela está próximo do seu limite máximo:

-É preferível utilizar encadeamento (*separate chaining*).

O método de percurso de grafos não dirigidos e não ponderados baseado na profundidade (*depth-first search*):

-Permite determinar a existência de ciclos.

Um grafo ponderado (não dirigido):

-Pode ter mais do que uma MST.

Um retalhista tem a informação dos seus produtos armazenada em suporte informático, utilizando uma lista que mantém ordenada pela chave de pesquisa a utilizar. Diga quais as questões a levar em consideração se pretender efetuar pesquisas baseadas na utilização de uma Árvore de Pesquisa Binária (BST).

-Se pretende efetuar pesquisas baseadas na utilização de uma BST é necessário introduzir-se a aleatoriedade no algoritmo (BST's Aleatórias). Desta forma, irá reduzir-se fortemente a probabilidade de ocorrência do caso pior. Também é necessário fazer uma amortização, ou seja um trabalho extra numa determinada ocasião, levando a garantir um limite superior eficaz do custo médio por operação. Por último, efetua-se uma otimização para garantir sempre a eficácia em todas as operações e acrescentar informação estrutural às BST's.

Em que situações julga preferível a utilização do método de percurso em profundidade DFS (*depth-first search*) em relação ao método de percurso em abrangência BFS (*breadth-first search*)?

-O método de percurso DFS tenta sempre ir mais fundo explorando todas as ligações dos vértices da árvore construída a partir das ligações de um vértice e se existirem vértices por visitar, recomeça o processo utilizando um desses vértices. Desta forma, o DFS é uma boa escolha quando se pretende visitar todos os vértices da árvore.

O método de percurso BFS explora uniformemente a partir de um vértice de partida, visita todos os vértices adjacentes ao vértice de partida, estabelecendo um conjunto de vértices e utiliza cada elemento deste conjunto para o ponto de partida. Desta forma, o BFS é uma boa escolha quando se pretende procurar o caminho mais curto.

Num grafo ponderado, a árvore MST (minimum spanning tree) corresponde ao caminho mais curto entre os vértices?

-A MST é uma árvore abrangente de custo mínimo de um grafo ponderado, ou seja é uma árvore abrangente cujo custo (soma do custo das ligações) é menor ou igual ao custo de qualquer outra árvore abrangente desse grafo.

Diga porque se costuma assumir que o método de Prim para a determinação da Árvore Abrangente de Custo Mínimo (*minimum spanning tree - MST*) de um grafo ponderado é adequado para grafos densos? (utilize a análise assimptótica).

-Porque o Algoritmo de Prim baseia-se na ideia de manter um corte do grafo, dividindo-o em vértices da MST e vértices não pertencentes à MST. O algoritmo de Prim pode ser considerando uma "extensão" do Algoritmo BFS (utiliza uma fila de espera com prioridade).

Diga se existirá alguma situação em que o caminho mais curto entre dois vértices de uma rede (grafo ponderado e dirigido) não seja único?

-O caminho mais curto entre dois vértices (s e t, por exemplo) é o caminho orientado do vértice s para t cujo custo não seja maior do que o custo de qualquer outro caminho de s para t. Desta forma, o caminho mais curto pode não ser único se existirem mais do que um caminho com o mesmo custo.

Numa tabela de endereçamento cada chave está associada a:

-A uma posição da tabela.

O método de Prim de determinação da MST em grafos ponderados:

-Utiliza uma fila de espera com prioridades.

O algoritmo de Dijkstra:

-Pode ser aplicado a grafos com pesos negativos nas ligações????

Nos algoritmos de optimização por colónia de formigas, o papel da feromona é:

-Reforçar os bons caminhos.

Ao utilizar tabelas de endereçamento com funções de *hash* para efectuar pesquisas em conjuntos de dados, sabemos que os mapeamentos possíveis vão ser quase todos utilizados. Diga se, neste caso, a resolução das colisões através de endereçamento aberto é a melhor opção.

-O endereçamento é uma boa opção mas faz um uso pouco eficaz da memória devido à memória que é reservada. Desta forma, a melhor opção é através do encadeamento pois não tem memória ocupada quando se pretende inserir mais elementos.

Diga como faria para determinar o caminho mais curto entre dois vértices de um grafo ponderado (mas não dirigido).

-Utilizaria o algoritmo de Dijkstra que funciona visitando os vértices no grafo a partir de um ponto de partida. Examina respetivamente o vértice mais próximo da origem que ainda não foi examinado, adicionando os seus vértices ao conjunto de vértices a serem examinados (expande-se a partir do ponto de partida até que atinja o objetivo). Desta forma, garante que encontra o caminho mais curto de um ponto de partida para um outro vértice, desde que nenhuma ligação tenha custo negativo.

A rede de estradas de uma pequena vila é modelada através de um grafo dirigido e ponderado. Como faria para saber se partindo de um determinado ponto conseguiria voltar ao mesmo local?

-Esta resposta vem baseada na classificação de ligações numa árvore DFS... Se, na criação da árvore DFS para a vila, encontrarmos back edges, então garantimos que existe um ciclo composto por essa ligação. De modo oposto, se não existir nenhuma back edge na árvore, então não seremos capazes de retornar ao ponto de partida.

Em que situações é que se deve utilizar o algoritmo de *Dijkstra* em vez do algoritmo A* para a determinação do caminho mais curto entre dois vértices de um grafo dirigido e ponderado?

-Deve-se utilizar o algoritmo de Dijkstra sempre que possível pois é o mais correto. Quando são vários elementos, este algoritmo fica demasiado lento o que leva a ser mais favorável o uso do algoritmo A* pois faz uso de heurísticas que leva à diminuição do número de vértices, tornando-o assim mais rápido. Desta forma, deve-se preferir o Dijkstra ao A* quando são poucos vértices.

Diga em que consiste o elitismo num algoritmo genético e que vantagem (ou vantagens) pode o algoritmo genético tirar desta técnica.

-O elitismo garante a passagem da melhor solução, ou o conjunto das melhores soluções, para a próxima geração. O elitismo garante que o algoritmo NÃO vá "perder tempo" a procurar soluções já eliminadas, impedindo qualquer tipo de "desprogresso".

O método de percurso de grafos não dirigidos e não ponderados baseado na abrangência (*breadth-first search*):

- Permite determinar caminhos mais curtos.

O método de Kruskal de determinação da MST em grafos ponderados:

- Depende da ordenação do custo das ligações.