DATA DE ATUALIZAÇÃO:
Sáb Ago 15 10:05:49 BRT 2015

CUSTO DE UM ALGORITMO E RECURSÃO:

- Exercício
  Entre n pessoas, uma celebridade é definida como sendo uma pessoa que é conhecida por todos, mas não conhece ninguém. Descreva um algoritmo que fazendo O(n) perguntas da forma "Você conhece x?" encontra, se existir, uma celebridade em um grupo com n pessoas.

- Exercício
  Considere o seguinte algoritmo que recebe como entrada um vetor A[0,..,n-1] de números inteiros:

```
int Caixa-Preta(int A[], int n) {
  int k=0, i=1, j=0;
  int h;
  while (i<n && k+j+1<n ) {
      if (A[k+j] == A[(i+j)%n]) {
          j = j+1;
      }
      else if (A[k+j] < A[(i+j)%n]) {
          i = i+j+1;
          j = 0;
      }
      else if (A[k+j] > A[(i+j)%n]) {
          h = max(i, k+j+1);
          k = h;
          i = h+1;
          j = 0;
      }
  }
  return k;
}
```

  Qual o consumo de tempo desse algoritmo? Expresse sua resposta em termos de notação O, mas procure dar a resposta mais justa possível. Justifique sua resposta.

- Exercício
  Uma função recursiva é aquela que se chama a si mesma (obrigatoriamente)?

- Exercício
  Dada uma matriz Anxm de inteiros considere os seguintes algoritmos baseados em comparações para encontrar o maior elemento:
  (a) Acha-se o máximo de cada linha, armazenando-se os resultados em um vetor, e depois acha-se o máximo do vetor.
  (b) Acha-se o máximo de cada coluna, armazenando-se os resultados em um vetor, e depois acha-se o máximo do vetor.
  supondo-se que n<m, qual deles realiza menos comparações? Justifique.

- Exercício
  Na seguinte função preencha os "??" corretamente para que o algoritmo permita fazer a busca de um elemento x em um vetor V[0,..., n-1].

```
int buscabinaria( int x, int n, int v[]) {
    int m, e = ??,  d = ??;
    while (e ?? d-1) {
        m = (e + d)/2;
        if (v[m] ?? x) e = m;
        else d = m; }
    return ??;
}
```

- Exercício
  Indique a ordem de crescimento (em função de N) dos tempos de execução de cada
  um dos seguintes procedimentos. Dica: Examine o número de vezes que a variável sum é atualizada.

```
int A1 (int N) {
    int i, j, sum=0;
    for (i=1; i<=N; i*=2)
        for(j=0; j<i; j++)
            sum++;
   return sum;
}

int A2 (int N) {
    int i, j, sum=0;
    for (i=1; i<=N; i*=4)
        for(j=0; j<i; j++)
            sum++;
   return sum;
}

int B1 (int N) {
    int n, i, sum=0;
    for (n=N; n>0; n/=2)
        for (i=0; i<n; i++)
            sum++;
    return sum;
}

int B1 (int N) {
    int n, i, sum=0;
    for (n=N; n>0; n/=4)
        for (i=0; i<n; i++)
            sum++;
    return sum;
}
```

- Exercício
  Uma empresa deseja comprar um software para resolver sistemas lineares. Existem
  no mercado quatro softwares para tal tarefa com custos operacionais
  diferentes. Seja n o tamanho da entrada para o sistema.
  -- ExtremeLS O(5(n^3)/2 + 2n/3),
  -- LUmaster O((n^3)/3 - (n^2)/4 + n/3),
  -- superQR O(2(n^3)/3 + n/2 - n/4),
  -- DonkeyShot O((n^4)/9 + n).
  Qual deles você escolheria? Justifique sua resposta.

- Exercício
  O elemento minimax de uma matriz é o menor elemento da linha que contém o
  maior elemento de uma matriz. Desenvolva um algoritmo para encontrar o elemento
  minimax

- Exercício
  O algoritmo abaixo está correto ou errado? Se estiver correto, mostre por meio
  de um exemplo sua corretude. Se estiver errado identifique e corrija o erro.

```
int pesquisa(int * v, int l, int r, int x) {
    if (l==r)
        return l;
    else {
        m = l+(r-l+1)/2;
        if (x <= v[m])
            return pesquisa(v,l,m,x);
        else
            return pesquisa(v,m+1,r,x);
    }
}
```

TABELAS DE DISPERSÃO:

- Exercício
  Descreva dois mecanismos diferentes para resolver o problema de colisões de várias chaves em uma mesma posição da tabela, destacando as vantagens e desvantagem.

- Exercício
  O que é o fator de carga? O que poderia dissertar de uma estrutura que utiliza uma tabela de espalhamento com fator de carga igual a 10?

ORDENAÇÃO EM TEMPO LINEAR:

- Exercício
  (a) Esquematize a prova de que a ordenação de n inteiros toma tempo : Omega(n log(n))
  (b) Descreva um algoritmo que permita ordenar n número em tempo O(n)
  (c) O resultado em (b) contradiz (a)? Justifique

- Exercício
  O Professor Progresso encontrou um algoritmo de ordenação baseado em comparações que gasta tempo O(n log(sqrt(n)).
  Considerando-se o limitante inferior para o tempo de ordenação, isso é possível? Justifique.

- Exercício
  O Prof. Countinho apresentou o seguinte algoritmo que realiza a ordenação, na forma crescente, de números armazenados em um vetor. O Prof. afirma que o algoritmo: (a) é baseado na contagem de comparações, e (b) o custo computacional para uma entrada de tamanho n é O(n).
  Quais dessas afirmações estão corretas? Apresente elementos concretos que confirmem ou neguem as afirmações.
  Para o teste, pode considerar o vetor v={62, 31, 84, 96, 19, 47}, contendo n=6 elementos.

```
void CountinhoSort (int v[], int n) {
    int i, j, count[n], s[n];

    for (i=0; i<n; i++)
        count[i] = 0;

    for (i=0; i<n-1; i++) {
        for (j=i+1; j<n; j++) {
            if (v[i]<v[j])
                count[j] += 1;
            else
                count[i] += 1;
        }
    }
    for (i=0; i<n; i++)
        s[count[i]] = v[i];
    for (i=0; i<n; i++)
        v[i] = s[i];
}
```

- Exercício
  Todo algoritmo de ordenação em tempo linear visto em aula faz uso de estruturas auxiliares (além do vetor de entrada)? Justifique.

ORDENAÇÃO PARCIAL:

- Exercício
  A ordenação parcial é um caso particular de ordenação total. Qual é a estratégia utilizada na ordenação parcial? Qual é a ideia principal considerada no algoritmo de Seleção parcial.

- Exercício
  Porque o algoritmo da inserção parcial visto em aula não preserva todos os elementos do vetor? Escreva o algoritmo que permita ordenar apenas os k primeiros elementos do vetor e preserve todos seus elementos.

- Exercício
  O que é um Min-Heap? Quais são as principais características desta estrutura de dados? Qual a altura da árvore? A árvore considerada no heap é uma árvore binária completa ou cheia? Em que caso a árvore seria completa ou cheia?

- Exercício
  Escrever o algoritmo de QuickSort parcial. Qual sua complexidade computacional em termos de número de comparações entre elementos? Seu algoritmo é estável?

- Exercício
  Escreva um algoritmo em que, dado um vetor v[0,..,n-1] de n elementos, permita ordenar os elementos no intervalo v[a,..,b]. Para 0 <= a <= b <= n-1.
  Qual a complexidade computacional em termos de número de comparações entre elementos?

- Exercício
  Todo algoritmo de ordenação parcial visto em aula faz uso de estruturas auxiliares? Em que casos a ordenação parcial não deve ser aplicada?

ÁRVORES BINÁRIAS:

- Exercício 3.1 [1]
  Desenvolver um algoritmo para produzir uma representação de uma árvore segundo o método de barras, dada a representação hierárquica.

- Exercício 3.10 [1]
  Justificar o motivo pelo qual uma árvore binária não é formalmente uma árvore.

- Exercício
  Provar ou dar contraexemplo: As árvores binárias são exatamente as árvores em que cada nó possui sempre 2 filhos.

- Exercício 3.24 [1]
  Quantos campos iguais a NULL possui a estrutura de armazenamento de uma árvore binária qualquer?

- Exercício 3.25 [1]
  Escrever um algoritmo para determinar o número de nós das subárvores de v, para cada nó v de uma árvore binária.

- Exercício 3.27 [1]
  Escrever um algoritmo para percorrer em nível uma árvore binária. Sugestão: utilizar uma fila.

- Exercício 3.31 [1]
  O percurso de uma árvore em ordem r-e-d resultou na impressão da sequência A, B, C, F, H, D, L, M, P, E, G, I, e o percurso da mesma árvore em ordem e-r-d resultou em F, C, H, B, D, L, P, M, N, A, I, G, E. Construa uma árvore que satisfaça esses percursos. Ela é única?

- Exercício [2]
  Desenhe uma árvore binária que com 17 nós que tenha a menor altura possível. Repita com a maior altura possível.

- Exercício [2]
  Escreva uma função que preencha corretamente todos os campos pai de uma árvore binária.

- Exercício
  Suponha que os nós da árvore tem a seguinte estrutura

```
struct cel {
   int         chave;
   int         conteudo;
   struct cel *pai;
   struct cel *esq;
   struct cel *dir;
};
typedef struct cel no;
```

  Escreva uma função que receba o endereço de um nó x de uma árvore binária e encontre o endereço do nó anterior a x na ordem r-e-d. Calcule o número de comparações entre nós (no pior e no melhor caso).

  Escreva uma função que receba o endereço de um nó x de uma árvore binária e encontre o endereço do nó anterior a x na ordem e-d-r. Calcule o número de comparações entre nós (no pior e no melhor caso).

ÁRVORES BINÁRIAS DE BUSCA (ABB):

- Exercício [2]
  Escreva uma função que decida se uma dada árvore binária é ou não é de busca.

- Exercício [2]
  Suponha que x->esq->chave <= x->chave <= x->dir->chave para cada nó x de uma árvore binária. Essa árvore é de busca?

- Exercício [2]
  Escreva uma função min que encontre uma chave mínima em uma árvore de busca.
  Escreva uma função max que encontre uma chave máxima.

- Exercício [2]
  Há uma relação muito íntima entre árvores de busca e o algoritmo de busca binária num vetor. Qual é, exatamente, essa relação?

- Exercício [2]
  Suponha que as chaves 50, 30, 70, 20, 40, 60, 80, 15, 25, 35, 45, 36 são inseridas, nesta ordem, em uma árvore de busca inicialmente vazia. Desenhe a árvore que resulta.

- Exercício [2]
  Escreva um método recursivo que devolva o número de nós de uma árvore binária.

- Exercício [2]
  Suponha que as chaves de uma ABB são números inteiros entre 1 e 10. Quais das sequências abaixo não podem ser as sequências de chaves examinadas em uma busca pela chave 5?
  -- 10, 9, 8, 7, 6, 5
  -- 4, 10, 8, 6, 5
  -- 1, 10, 2, 9, 3, 8, 4, 7, 6, 5
  -- 2, 7, 3, 8, 4, 5
  -- 1, 2, 10, 4, 8, 5

- Exercício
  O que é uma árvore binária com costura? Em que casos seria útil este tipo de estruturas? Desenvolva um algoritmo para percorrer todos os elementos da árvore (no sentido e-r-d).

- Exercício
  Suponha que estamos procurando pela chave 363 em uma árvore binária de busca cujas chaves são número naturais entre 1 e 1000. Durante a busca, uma certa sequência de chaves é examinada. Quais das seguintes sequências são impossíveis? Justifique.

- Exercício
  Desenvolva um algoritmo para juntar duas ABBs. Qual é o custo computacional
  desse algoritmo? Seja ha e hb a altura das árvores binárias de busca Ta e Tb,
  respectivamente. Pode se afirmar que a altura da árvore resultante será, no
  minimo ha+hb? Justifique.

ÁRVORES AVL:

- Exercício 5.9 [1]
  Detalhar um algoritmo de exclusão (eliminação/remoção) de nós em árvores AVL.

- Exercício
  Suponha que serão realizadas as seguintes inserções de chaves na árvore AVL
  que contem a chave 10: {1,2,3,4,5,6,7}.
  -- Apresente a árvore AVL resultante (1 árvore)
  -- Indique o tipo de rotações consideradas em cada inserção.

- Exercício
  Escreva um método que devolva a menor altura de uma folha de uma árvore AVL.

- Exercício
  Um certo professor Amongus afirma que a ordem pela qual um conjunto fixo de
  elementos é inserido em uma árvore AVL não interessa - sempre resulta na mesma árvore.
  Apresente um pequeno exemplo que prove que ele está errado.

- Exercício
  Analise uma árvore T que armazena 100.000 itens. Quais são o pior e o melhor
  casos em relação à altura de T das seguintes árvores:
  -- T é uma árvore binária completa;
  -- T é uma árvore AVL.

- Exercício
  Seja uma árvore AVL T. Considere a inserção de um nó q em T, que tornou T
  desregulada. Seja p o nó desregulado mais próximo das folhas.
  -- Qual o valor exato de |h(p->esq) - h(p->dir)| ? Por que não pode ser nem mais nem menos?
  -- Supondo h(p->dir) > h(p->esq) então existe um filho direito u de p. Por que
  necessariamente temos |h(u->dir) - h(u->esq)| = 1? Por que não pode ser 2 ? Por que não
  pode ser 0 ?

- Exercício
  Suponha que os nós da árvore tem a seguinte estrutura

```
struct cel {
   int         chave;
   int         conteudo;
   struct cel *pai;
   struct cel *esq;
   struct cel *dir;
};
typedef struct cel no;
```

  Uma árvore é balanceada no sentido AVL se, para cada nó x, as alturas das
  subárvores que têm raízes x->esq e x->dir diferem de no máximo uma unidade.
  Escreva uma função que decida se uma dada árvore é balanceada no sentido AVL.
  Procure escrever sua função de modo que ela visite cada nó no máximo uma vez.

- Exercício
  Quais são as diferenças/semelhanças entre Árvores binárias, ABBs, AVLs, e estrutura de
  heap? Todos os algoritmos baseadas nessas estruturas tem complexidade
  proporcional a O(log(n)), onde n é o número de elementos da estrutura?

ÁRVORES RUBRO-NEGRAS:

- Exercício
  Qual é a altura de uma árvore rubro-negra? Esquematize a prova sobre a altura da árvore.

- Exercício
  Em linguagem C/C++ especifique as características mínimas que deve ter um nó.
  Defina a estrutura (e.g., struct).

- Exercício
  O que é uma árvore 2-3-4?

- Exercício
  Prova com contra-exemplo:
  - Toda árvore AVL é uma árvore Rubro-Negra.
  - Toda árvore Rubro-Negra é uma árvore AVL.

- Exercício:
  Implemente um programa que calcule a altura mínima de uma árvore rubro-negra.

- Exercício
  Compare as árvores ABB, AVL e Rubro-Negras, considerando:
  - Consulta;
  - Inserção;
  - Balanceamento.

- Exercício
  Crie uma função para, dada uma árvore binária, verifique se esta é rubro-negra.
  Considere a seguinte estrutura:

```
struct cel {
    int chave;
    int conteudo;
    struct cel *filho[2];
    char cor;
};
typedef struct cel no;
```

  Assinatura: `int verificarArvoreRN(no *raiz)`

- Exercício
  Prove ou dê contra-exemplo: seja uma árvore rubro-negra cuja raiz possui a cor
  rubra. Se esta for alterada para negra, a árvore mantém-se rubro-negra.

ÁRVORES TRIE:

- Exercício
  O que é uma árvore Trie? Indique suas principais características.

- Exercício
  Descreva, em pseudocódigo, o algoritmo de inserção em uma árvore Trie.

- Exercício
  Dadas as palavras abaixo, crie uma árvore Trie (graficamente) para armazená-las.
  roupa, rato, casa, castor, mesa, morro, gorro, galho.

- Exercício
  A partir de um prefixo p fornecido, escreva uma função que imprima
  até 10 sugestões de chaves válidas (isto é, contidas na árvore Trie) que
  tenham p como prefixo. Forneça as palavras em ordem alfabética

  Considere a seguinte estrutura:

```
#define TAMANHO_ALFABETO (27)
#define CHAR_TO_INDEX(c) ((int)c - (int)'a')
struct trie_cel {
    char tipo; // 'I': interno / 'P': palavra
    struct trie_cel *filho[TAMANHO_ALFABETO];
};
typedef struct trie_cel no;
```

  Assinatura: `void imprimirSugestoes(char *p, no *raiz)`

- Exercício:
  Dê um exemplo de caso de uso de árvores Tries.

ÁRVORES PATRICIA:

- Exercício
  O que é uma árvore Patricia? Indique suas principais características.

- Exercício
  Em que casos a árvore Patricia é menos vantajosa a árvore Trie?

- Exercício:
  Dê um exemplo de caso de uso de árvores Patricia.

ÁRVORES B:

- Exercício
  O que é uma árvore paginada?

- Exercício
  O que é uma árvore B? Indique suas principais características.
  Todas as chaves aparecem no nível das folhas?

- Exercício
  Qual é a diferença entre árvore B, B+, B*? Apresente informações detalhadas.

- Exercício
  Descreva em pseudocódigo o algoritmo para inserir uma nova chave em uma árvore B.

- Exercício
  Explique as vantagens de uso de árvores B em relação à busca de dados em
  memória secundária (discos rígidos).

- Exercício
  Desenhe uma árvore B de ordem 2 que contenha as seguintes chaves: 1, 3, 6, 8,
  14, 32, 36, 38, 39, 41, 43 (inserir nessa ordem).

- Exercício
  Mostre os resultados de inserir as chaves a seguir em uma árvore B de ordem 3 inicialmente vazia:
  F, S, Q, K, C, L, H, T, V, W, M, R, N, P, A, B, X, Y, D, Z.

CONJUNTOS DISJUNTOS:

- Exercício
  Como é definida a função inversa de Ackerman (considere apenas uma variável)?

- Exercício
  Em que situações algoritmos union-find são utilizados e quais as operações
  fundamentais nesta estrutura?

- Exercício
  Discuta, em pseudocódigo, duas formas de implementação de algoritmos
  union-find, destacando vantagens e desvantagens.

- Exercício
  Explique a heurística de união ponderada.

- Exercício
  Explique o problema de árvore geradora mínima, dê exemplos e aplicações
  relacionadas.

- Exercício
  Mostre a estrutura de dados resultante e as respostas devolvidas pelas
  operações Find_Set no seguinte programa, usando
  a) lista ligada com heurística de união ponderada.
  b) floresta de conjuntos disjuntos com união ponderada e compressão de caminhos.
  Assuma que, caso os conjuntos contendo i e j sejam do mesmo tamanho, Union(i,j)
  concatena a lista de j na lista de i.

```
for (i = 1; i <= 16; i++)
    Make_Set(i);
for (i = 1; i <= 15; i = i + 2)
    Union(i,i+1);
for (i = 1; i <= 13; i = i + 4)
    Union(i,i+2);
Find_Set(2);
Find_Set(9);
```

ORDENAÇÃO EXTERNA:

- Exercício
  O que é ordenação externa?

- Exercício
  Quais algoritmos de ordenção vimos em aula?

REFERÊNCIAS:
---

[1] SZWARCFITER, J. L.; MARKEZON, L. Estruturas de Dados e seus Algoritmos, 3a edição, LTC, 2010.
[2] FEOFILOFF, P. Projeto de Algoritmos em C. http://www.ime.usp.br/~pf/algoritmos/