UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
CENTRO MULTIDISCIPLINAR PAU DOS FERROS – CMPF
INTERDISCIPLINAR EM TECNOLOGIA DA INFORMAÇÃO

MATEUS VICTOR DA CUNHA TOMÁS

LABORATÓRIO DE ALGORITMOS E ESTRUTURAS DE DADOS II

Pau dos Ferros – 2024

Q01

fib(8) => fib(7)+fib(6)

fib(7) = > fib(6)+fib(5)

fib(6) => fib(5)+fib(4)

fib(5) => fib(4)+fib(3)

fib(4) => fib(3)+fib(2)

fib(3) => fib(2)+fib(1)

fib(6) => fib(5)+fib(4)

fib(5) => fib(4)+fib(3)

fib(4) => fib(3)+fib(2)

fib(3) => fib(2)+fib(1)

é calculado fib(4) 5 vezes

Q02

Se quisermos calcular o Fibonacci de um número n maior do que 4, podemos simplesmente continuar a sequência até chegarmos ao número n. Por exemplo, se quisermos calcular o Fibonacci de 7, continuamos a sequência até o sétimo número (excluindo o zero inicial), que é 13.

Q03

```
#include <stdio.h>

void fibonacci(int n, int fib[]) {
    fib[0] = 0;
    fib[1] = 1;
    for (int i = 2; i <= n; i++) {
        fib[i] = fib[i-1] + fib[i-2];
    }
}

int vezes_fibonacci_calculado(int n, int m) {
    if (m < n) {
        return 0;
    }
    int fib[m+1];
```

```
    fibonacci(m, fib);
    return fib[n];
}
int main() {
    int n = 5;
    int m = 10;
     printf("Fibonacci de %d é calculado %d vezes ao calcular Fibonacci de %d\n", n, vezes_fibonacci_calculado(n, m), m);
    return 0;
}
```

Q04

Q05

Q06

Q07

Q08

Q09

Q10

Q11

Q12

Q13

Q14

Q15

Q16

Q17

Q18

Q19

Q20

Q21

Q22

(a) n^2 é $\Theta(n^3)$; Falso. A notação Θ é usada para descrever um limite superior e inferior assintótico estrito. n^2 não é $\Theta(n^3)$ porque n^2 cresce a uma taxa significativamente mais lenta do que n^3 quando n se aproxima do infinito.

(b) n^3 é $\Theta(n^2)$; Falso. Similarmente, n^3 não é $\Theta(n^2)$ porque n^3 cresce a uma taxa significativamente mais rápida do que n^2 quando n se aproxima do infinito.

c) $\log10(n^2)$ é $\Theta(\lg(n))$ Verdadeiro. Primeiro, note que $\log10(n^2) = 2 * \log10(n)$. Além disso, a mudança de base do logaritmo não afeta a taxa de crescimento assintótico, então $\log10(n)$ é $\Theta(\lg(n))$. Portanto, $2 * \log10(n)$ também é $\Theta(\lg(n))$.(

d) $n^2 \cos^2(n)$ é $\Theta(n^2)$ Verdadeiro. A função $\cos^2(n)$ oscila entre 0 e 1 para todos os valores de
n. Portanto, $n^2 \cos^2(n)$ é limitado acima por n^2 e abaixo por 0. Isso significa que $n^2 \cos^2(n)$ é $\Theta(n^2)$.

Q23

(a) n^2 é $\Omega(n^3)$; Falso. A notação Ω é usada para descrever um limite inferior assintótico. n^2 não é $\Omega(n^3)$ porque n^2 cresce a uma taxa significativamente mais lenta do que n^3 quando n se aproxima do infinito.

(b) n^3 é $\Omega(n^2)$; Verdadeiro. n^3 é $\Omega(n^2)$ porque n^3 cresce a uma taxa significativamente mais rápida do que n^2 quando n se aproxima do infinito.

c) $\log10(n^2)$ é $\Omega(\lg(n))$ Verdadeiro. Primeiro, note que $\log10(n^2) = 2 * \log10(n)$. Além disso, a mudança de base do logaritmo não afeta a taxa de crescimento assintótico, então $\log10(n)$ é $\Omega(\lg(n))$. Portanto, $2 * \log10(n)$ também é $\Omega(\lg(n))$.

d) $n^2 \cos^2(n)$ é $\Omega(n^2)$ Falso. A função $\cos^2(n)$ oscila entre 0 e 1 para todos os valores de n. Portanto, $n^2 \cos^2(n)$ não é $\Omega(n^2)$ porque não há um valor constante c tal que $n^2 \cos^2(n)$ seja sempre maior ou igual a $c * n^2$ para todos os n suficientemente grandes.

Q24

(a) $c1O(f(n)) = O(c1 * f(n))$ Verdadeiro. A notação O ignora constantes multiplicativas. Portanto, $c1O(f(n))$ é o mesmo que $O(c1 * f(n))$.

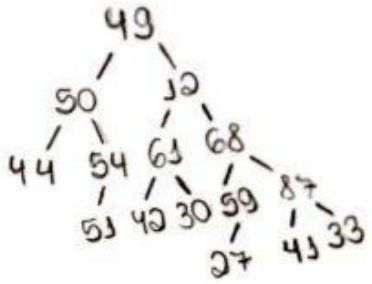
(b) $O(f(n) + g(n)) = O(f(n) * g(n))$ Falso. A soma de duas funções não é necessariamente a mesma que o produto das duas funções. Por exemplo, se $f(n) = n$ e $g(n) = n$, então $f(n) + g(n) = 2n$, mas $f(n) * g(n) = n^2$.

© $O(f(n)) + O(g(n)) = O(\max(f(n), g(n)))$ Verdadeiro. A soma de duas funções é dominada pela função que cresce mais rapidamente. Portanto, $O(f(n)) + O(g(n))$ é o mesmo que $O(\max(f(n), g(n)))$.

(d) f(n)O(g(n)) = O(g(n)) Falso. Multiplicar uma função por outra função pode alterar a taxa de crescimento da função resultante. Portanto, f(n)O(g(n)) não é necessariamente o mesmo que O(g(n)).

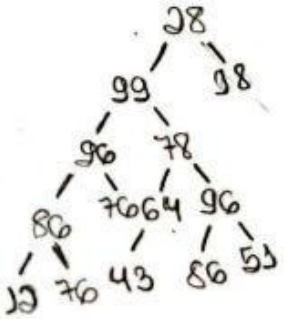
Questão 25:

(49 - 50 - 44 - 54 - 13 - 61 - 68 - 87 - 59 - 30 - 43 - 51 - 33 - 43 - 27)



Questão 26:

(28 - 99 - 78 - 96 - 64 - 63 - 51 - 86 - 43 - 76 - 86 - 76 - 13 - 18 - 89)

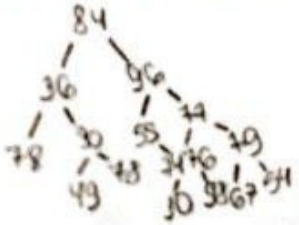


Questão 27:

(61 - 66 - 83 - 39 - 78 - 95 - 19 - 63 - 45 - 44 - 55 - 47 - 45 - 86)

Questão 28:

(84 - 96 - 36 - 30 - 78 - 49 - 77 - 55 - 79 - 76 - 74 - 54 - 67 - 88 - 10)



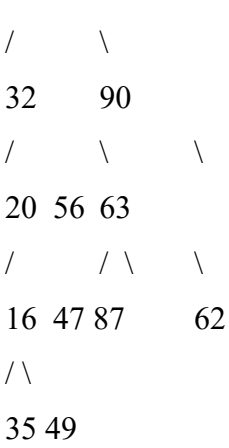
Q25

Q26

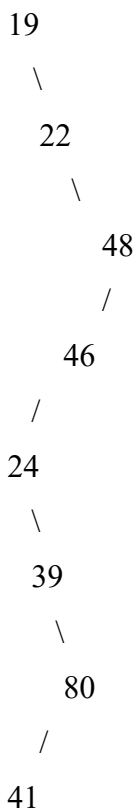
Q27

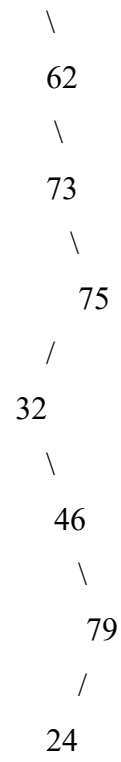
Q28

Q29 Construa uma árvore binária de busca com os elementos: 91 32 56 90 63 49 20 62 47 87 16 56 35 32 90



Q30 Construa uma árvore binária de busca Zigue-Zague com os elementos: 19 22 48 46 24 39 80 41 62 73 75 32 46 79 24





Utilizem das questões Q25 a Q30 para responder as questões Q31 a Q36:

Q31 Calcule os sucessores e antecessores dos nós raízes. 1º árvore: antecessor - 44; sucessor - 50

2º árvore: antecessor - 18; sucessor - 43 3º árvore: antecessor - 55; sucessor - 63 4º árvore: antecessor - 79; sucessor - 96 5º árvore: antecessor - 90

6º árvore: sucessor - 22

Q32 Localize o nível de todos os nós. 1º árvore: 5 níveis

2º árvore: 5 níveis 3º árvore: 4 níveis 4º árvore: 5 níveis 5º árvore: 5 níveis 6º árvore: 15 níveis

Q33 Localize a altura de todos os nós. 1º árvore: 4

2º árvore: 4

3º árvore: 3

4º árvore: 4

5º árvore: 3

6º árvore: 14

Q34 Realize o percurso em pré-ordem.

1º árvore: Pré-ordem: 49-50-44-54-51-12-61-42-30-68-59-27-87-41-33

2º árvore: Pré-ordem: 28-99-96-86-12-76-76-78-64-43-98-86-51-18

3º árvore: Pré-ordem: 61-83-19-86-95-45-47-66-78-55-44-39-45-63

4º árvore: Pré-ordem: 84-36-78-30-49-79-96-55-74-10-77-76-98-79-67-54

5º árvore: Pré-ordem: 91-32-20-16-56-47-35-49-87-90-63-62

6º árvore: Pré-ordem:19-22-48-46-24-39-80-41-62-73-75-32-46-79-24

Q35 Realize o percurso em pós-ordem

1º árvore: Pos-ordem: 44-51-54-50-42-30-61-27-59-41-33-87-68-12-49

2º árvore: Pos-ordem: 12-76-86-76-96-43-64-86-51-96-78-99-18-28

3º árvore: Pos-ordem: 86-19-45-47-95-83-55-44-78-45-63-39-66-61

4º árvore: Pos-ordem: 78-49-79-30-36-10-74-55-98-76-67-54-79-77-96-84

5º árvore: Pos-ordem: 16-20-35-49-47-87-56-32-62-63-90-91

6º árvore: Pos-ordem:24-79-46-32-75-73-62-41-80-39-24-46-48-22-19

Q36 Realize o percurso em in-ordem.

16-20-32-35-47-49-56-87-91-90-63-62

19-22-48-46-24-39-80-41-62-73-75-32-46-79-24

Q37

a)verdadeiro

b)falso

c)falso

d)verdadeiro

e)Falso

g)falso

Q38 Qual a altura de uma árvore cheia que possui N=324 nós?

h=altura

N =2^h-1

DadoB =324

$h = \log_2 (324+1)$

$h = \log_2 (325)$ = aproximadamente 8,34

Q39 Quantos nós faltam para uma árvore cheia com N=348 nos se torne uma árvore completa?

Calcule a altura da árvore com 348

h =log2(348+1)=log2(349)

h≈8.45

A altura precisa ser um número inteiro, e dado que a altura não é um inteiro, arredondamos para cima para encontrar a altura mínima necessária para uma árvore completa que poderia acomodar 348 nós:

hmin=⌈8.45⌉=9

Calculamos o número total de nós para uma árvore completa com altura hmin=9:

Ntotal=2hmin−1=29−1=511

Subtraímos o número atual de nós (348) do número total de nós necessário para uma árvore completa com essa altura:

Nfaltando=511−348=163

Portanto, faltam 163 nós para que uma árvore cheia com 348 nós se torne uma árvore completa (no sentido de ter uma altura que permitiria um último nível completamente preenchido).

Q40 Quantos nós precisariam ser removidos para que uma árvore cheia com N=538n´ os se torne uma árvore completa?

Para determinar quantos nós precisariam ser removidos para que uma árvore cheia com N=538 nós se torne uma árvore completa, precisamos entender que uma árvore cheia já é completa por definição. Em uma árvore cheia, todos os níveis estão completamente preenchidos. Portanto, não é possível remover nós de uma árvore cheia para torná-la completa, pois ela já possui o número máximo de nós possível para sua altura.

Uma árvore cheia é aquela em que cada nível é completamente preenchido, exceto talvez o último, onde os nós devem ser tão à esquerda quanto possível. Além disso, o número total de nós em uma árvore cheia é dado por 2h−1 onde *h*h é a altura da árvore.

Para N=538 nós, precisamos calcular a altura da árvore cheia correspondente:

h=log2(538+1)=log2(539)

h≈9.10h≈9.10

Como a altura da árvore precisa ser um número inteiro, arredondamos para cima para encontrar a altura mínima necessária para uma árvore cheia que poderia acomodar 538 nós:

hmin=⌈9.10⌉=10

Agora, podemos calcular o número total de nós para uma árvore cheia com altura hmin=10:

Ntotal=2hmin−1=210−1=1023

Dado que N=538 e N_{total}=1023N_{total}=1023 não é possível remover nós de uma árvore cheia para torná-la completa. A árvore cheia já possui o número máximo de nós possível para sua altura e estrutura. Portanto, não seria necessário remover nenhum nó para atingir a condição de uma árvore completa.

Q41 Qual a maior altura do nó raiz em uma árvore estritamente binária com N=251nós.
Para encontrar a maior altura possível de uma árvore estritamente binária com N=251N=251 nós, podemos usar a fórmula que relaciona o número de nós (N) com a altura da árvore (*h*):
N=2^(h+1) −1
Dado que N=251N=251, podemos resolver essa equação para *h*:
N =2^(h+1)-1
Adicionando 1 em ambos os lados e resolvendo para *h*, temos:
252=2h+1
28=2*h*+1
8=*h*+1
h=7
Portanto, a maior altura possível de uma árvore estritamente binária com N=251 nós é *h*=7 .