Grupo 1 – Complexidade Algorítmica (20%)  
  
1.1 Para os pares de funções seguintes, escreva na caixa em branco Θ, O ou Ω para tornar a armação verdadeira.

Note que se as funções em causa tiverem uma relação Θ, não deverá escrever O ou Ω.  
  
a) 3n \_ n^3  
b) n raiz(n) \_ nlogn

c) n^n \_ n!

d) 2n\_(40n)2^(2n+2)\_2^n

1.2 Descreva o pior caso no tempo de execução para os seguintes pedaços de código, usando notação O para a variável n. Use o limite mais "apertado" possível (ex: indicar todos como sendo O(n!) resultará em 0 pontos).  
  
a) for (int i = 1 ; i<n; i=i\*3)  
 for (int j = n; j>0 ; j=j-3)

count++;  
  
b) for (int i = 0; i<n; i++)  
 for (int j = 20; j>=0 ; j--)

count++;  
  
c) int f1 (int n){

if (n <= 0) return 1;

return f1(n/3) + 3;

}  
  
d) int f2(int n, int [] v){

if (n <= 0) return v[0];

return 1 + f2(n/2,v) + f2(n/2,v);  
 }  
  
1.3 Justique a sua resposta para a alínea (d) da pergunta anterior, escrevendo a recorrência e desenhando a respectiva árvore de recorrência, explicando porque dá origem à complexidade indicada.  
  
1.4 Programa Complexidade Tempo para n1 = 10 Tempo para n2 = 20  
 A Θ (n^3) 40 ...

B Θ (log n) 40 ...

C Θ (1) ... 40

1.5 Suponha que está a criar uma estrutura de dados para representar conjuntos de números inteiros. Indique

a complexidade (notação O do pior caso) do melhor algoritmo que consegue imaginar para a implementação

de alguns métodos. Suponha que o conjunto tem n inteiros entre 1 e k e justique sucintamente as suas respostas.  
  
a) Complexidade temporal para saber se o conjunto contém um elemento v usando:  
  
Array Ordenado:  
Array de booleanos:   
  
b) Complexidade temporal para remover um elemento no conjunto usando:  
  
Array Ordenado:  
Array de booleanos:  
  
c) Complexidade espacial para armazenar o conjunto usando:  
  
Array Ordenado:  
Array de booleanos:  
  
Grupo 2 – Árvores (20%)

A black and white diagram

Description automatically generated  
  
2.1 Considerando a árvore da gura seguinte, responda às seguintes alíneas:  
  
a) Os nós da árvore em inorder:

b) Os nós da árvore em preorder:

c) Os nós numa pesquisa em largura:

2.2 Considere uma classe BTree<T> de uma árvore binária genérica tal como dada nas aulas com um atributo root a apontar para a raíz que é um BTNode<T> representando um nó com atributos value, left e right.  
  
a) Implemente um método levels() da classe BTree<T> que devolve o número de níveis de profundidade da árvore (por exemplo, a árvore da pergunta 2.1 tem 4 níveis).

Não pode usar qualquer outro método já existente da classe, mas pode usar métodos auxi-

liares. Indique a complexidade temporal do método que descreveu.  
  
  
  
b) Implemente um método estático maxs(BTree<Integer> t) que devolve um array com o maior elemento de cada um dos níveis da árvore contendo em cada posição i o maximo desse nível. Por exemplo, para a árvore da pergunta 2.1, deveria ser devolvido um array

contendo [A, max(C, D), max(B, G, F), E].

Não pode usar qualquer outro método já existente da classe, excepto o método levels() da pergunta anterior, mas pode criar outros métodos auxiliares. Indique a complexidade tem-

poral e espacial do método que descreveu.  
  
Grupo 3 – Árvores Binárias de Pesquisa (20%)  
  
3.1 Em termos gerais qual a diferença de uma árvore binária e uma árvore binária de pesquisa? Jusitifica a resposta  
  
3.2 Insere por esta ordem numa árvore binária de pesquisa 45,42,43,47,46,48  
  
a) Como fica a árvore original após a inserção dos 6 números?  
  
b) Como fica a árvore (original) após a remoção do número 45?   
  
c) Supondo que quer inserir 10 letras minúsculas entre (a – j) numa árvore binária de pesquisa.  
  
 1) Qual a menor altura possível da árvore ?

2) Indique a ordem que inseria os elementos para garantir a altura mínima descrita anteriormente.  
 \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ (10 espaços para colocar as letras)  
  
d) Imagina uma árvore binária de pesquisa com altura 40 (40+1 niveis) portanto 41 niveis.  
  
 1) Qual o maior número possível de nós que pode ter ? Justifica a resposta  
 2) Qual o menor número possível de nós que pode ter ? Justifica a resposta  
  
Grupo 4 - Filas de Prioridade e Heaps (20%)  
  
4.1 Imagine que tem uma maxHeap descrita pelo seguinte array  
  
A black line with numbers

Description automatically generated  
  
Para as 3 alíneas de baixo desenhe a árvore (a heap) correspondente (sempre já com o invariante reposto):  
  
a) Qual a heap representada pelo array original?  
b) Como fica a heap (original) depois de adicionarmos um 8?  
c) Como fica a heap (original) depois de remover o máximo?  
  
4.2 Explique como poderia usar uma maxHeap para ordenar um conjunto de n números por ordem crescente, indicando que operações usava e qual seria a complexidade temporal e espacial do seu algoritmo.  
  
Grupo 5 – Completar Blocos de Código  
  
5.1 Recursividade  
  
void sets (int v[] )

void goSets (int cur, int v [], Boolean used[] )

5.2 BTree <T>

boolean contains( T value)  
void print DFS ()

5.3 BSTree <T>

boolean remove (T value)  
(…)

BTSNode<T> remove(BTSNode<T> n, T value)  
  
5.4 MinHeap<T>  
  
T removeMin()  
void downHeap(int i)