Giullio Emmanuel da Cruz Di Gerolamo

RA: 790965

Prova P2

Diagram

Description automatically generated

**Resposta:**

Node\* buscar(Node\*R, int Ch){

while(R){

if(Ch < R->chave)

R = R->esq;

else if(Ch > R->chave)

R = R->dir;

else

return R;

}

return NULL;

}

void insere(Node \*\*R, int Ch){

if(\*R == NULL){

\*R = malloc(sizeof(Node));

(\*R)->chave = Ch;

(\*R)->esq = NULL;

(\*R)->dir = NULL;

}

else{

if(Ch < (\*R)->chave)

insere(&((\*R)->esq), Ch);

else

insere(&((\*R)->dir), Ch);

}

}

**Na implementação(main):**

Node \*busca = NULL;

busca = buscar(R, Ch);

if(busca)

printf("\n\tValor já se encontra na arvore");

else

insere(&R, Ch);

Graphical user interface, text, application, email

Description automatically generated

**Resposta:**

**a)**

int GetAltura(Node \*R){

if(R == NULL){

return -1;

}

else{

int esquerda = GetAltura (R->esq);

int direita = GetAltura (R->dir);

if(esquerda > direita)

return esquerda + 1;

else

return direita + 1;

}

}

**b)**

O(n)

Pois todos os nós são acessados Nessa função.

Graphical user interface, text, letter, email

Description automatically generated

**Resposta:**

**a)**

void ordena(int\* V, int N)

{

int i, j;

for (i=N-1; i>0; i--)

int troca = 0;

for (j=0; j<i; j++){

if (V[j]>V[j+1]) {

int temp = V[j];

V[j] = V[j+1];

V[j+1] = temp;

troca = 1;

}

if (troca == 0)

return;

}

}

**b)**

O algoritmo de ordenação utilizado foi do tipo bolha ou troca (bubble sort), e a ordem de eficiência de tempo é O(n²).

Diagram, engineering drawing

Description automatically generated

**Resposta:**

bool isHeap(int \*V, int LastPosition)

{

int n = LastPosition;

for (int i=0; i&lt;=(n-2)/2; i++)

{

if (arr[2\*i +1] &gt; arr[i])

return false;

if (2\*i+2 &lt; n &amp;&amp; arr[2\*i+2] &gt; arr[i])

return false;

}

return true;

}