arvore binario com fio

estrutura derivada de arvore binaria em que ponteiros nulos são usados para para armazenar o endereco predecessor in ordem.

Arvores tem ligação direta com recursividade e recursividade é a base para a arvores binarias ,arvores essas que, tem no máximo, só dois filhos, como geralmente esse tipo de arvore trabalha com que esperam um ou dois elementos, é por isso que os nós da arvore tem no máximo dois filhos, pq um deles sempre vai ficar vazio para que o outro ocupe a posição.

Da pra definir uma arvore binário como sendo uma arvore vazia ou um nó raiz que contem duas sub-árvores, identificadas por sub da direita e sub da esquerda.

Notação textual para descrever arvores binarias:

< > arvore vazia;

< raiz dir e esq> arvore não vazia;

Com essa notação, a árvore da Figura 13.5 é representada por: <a<b<><d<><>>><c<e<><>><f<><>>>>

Por definição, essa porra dessa arvore binaria sempre vai sempre ser definida como esquerda ou direita de uma arvore maior (o começo do bagulho) e qualquer uma das duas sub-arvores podem ser nulas

A arvore tem n elementos, então tem n + 1 ponteiros, se uma arvore tem k = 1 elemento, então tem dois ponteiros nulos adicionando um elemento numa posição ai da arvore, o ponteiro deixa de ser nulo, mas tbm aparecem mais outros dois ponteiros nulos, pq novos elementos sempre são folhas, então a nova arvore tem (k +1 ) -1+2 = (k + 1) + 1 ponteiros vazios.

A ligação de fio se aproveita dessa memoria jogada fora para apontar pro predecessor ou sucessor se não tiver nenhum filho esquerda ou direita, tornando o percurso in-ordem mais facil e eficiente.

exemplo:

struct nodo {

int chave;

struct nodo \*esq;

struct nodo \*dir;

int eFlag, dFlag;

}

struct nodo \*predecessor (struct nodo \*p) {

struct nodo \*aux;

if(p→eFlag == 1)

aux = p→esq;

else {

aux = p→esq;

while(aux→dFlag == 0) {

aux = aux→dir;

}

}

return aux;

}

struct nodo \*sucessor(struct nodo \*p) {

struct nodo \*aux;

if(p→dFlag == 1)

aux = p→dir;

else {

aux = p→dir;

while(aux→eFlag == 0) {

aux = aux→esq;

}

}

return aux;

}

https://pt.scribd.com/document/26510953/Estrutura-de-dados-Arvore-binaria