TI DSP,Xilinx zynq FPGA,MCU 및 Xilinx zynq FPGA 프로그래밍 전문가 과정

강사-INNOVA LEE(이상훈)

Gccompil3r@gmail.com

학생-윤지완

Yoonjw7894@naver.com

1.enqueue 재귀함수 제거

```
#include<stdio.h>
#include<malloc.h>

typedef struct tree
{
int data;
struct tree *link;
}tree;


tree get_node()
{
tree *tmp;
tmp=(tree*)malloc(tree);
tmp->link=NULL;
return tmp;
}
void insert(treee **root,int data)
{
while(*root)
{
root=&(*root)->link

}
*root=get_node();
(*root)->data=data;
}

void print_queue(tree *root)
{
if(root)
printf("rootd의 값= %d",root->data);
}
void dequeue(tree *root,int data)
{
tree *tmp=root;
if(*root==NULL)
printf("no delete");
}
while(*tmp)
{
if((*tmp)->data!=data)
{
tmp=&(*tmp)->link;
}

else if((*tmp)->data==data)
{
free(tmp);
return root->link;
}
}
```

```
int main(void)
{
tree *root=NULL;
int data[4]={10,20,30};
int i;
for(i=0;data[i];i++)
insert(&root,data[i]);
print_queue(root);
delete(root,20)
return 0;
}
```

2.2진트리 재귀 함수 제거

```
#include<stdio.h>
#include<malloc.h>

typedef struct tree
{
int data;
struct tree *left;
struct tree *right;
}tree;

typedef struct   stack
{
void   data;
struct stack link;
}stack;


tree get_node()
{
tree *tmp;
tmp=(tree*)malloc(tree);
tmp->left=NULL;
tmp->right=NULL;

return tmp;
}
void inert(tree **root,int data)
{
tree **tmp=root;
stack *tmp=NULL;

while(*tmp)
{
if((*tmp)->data>data)
tmp=&(*tmp)->left;
else if((*tmp)->data<data)
tmp=&(*tmp)->right;
}
*tmp=get_node();
(*tmp)->data=data;
}
```

```
stack push(tree *root, int *data)
tree *tmp=root;
stack *top=NULL;
if(*top==NULL)
return ;
}}
*top=get_node()
(*top)->data=(tree*)malloc(void *);
(*top)->data=data;
(*top)->link=tmp;
}

stack pop(tree **top)
{
tree *tmp=*top;
void *data=NULL;
if(*top==NULL)
{
return NULL;
}
data=(*top)->data;
*top=(*top)->link;
return data;
}

bool stack is not empty(stack *top)
{
if((*top)!=NULL)

return true;
else
return false;
void Qpoint(tree **root,int data)
{
int counter;
tree **tmp=root;
stack *top=NULL;
push(&top,*tmp)
while(stack is not empty(top))
{
tree *t=(tree*)pop(&top);
tmp=&t;
if((*tmp)->left)
printf("left 의 값은 = %d",(*tmp)->data->left);
else if((*tmp->right)
printf("right 의 값은 = %d",(*tmp)->data->right);
push(&top,(*tmp)->right);
push(&top,(*tmp)->left);

}
void point(tree *root)
{

if(root)
printf("root=%d₩n",root->data);
else if(root->left)
```

```
printf("left = %d₩n",root->data->left);
else if(root->right)
printf("right= %d₩n",root->data->right);
point(root->left);
point(root-<left);
}
void delete(tree **root, int *data)
{
tree **tmp=root;
int num;
if(*root==NULL)
{
return ;
}
while(*tmp)
{
if((*tmp)->data>data)
tmp=&(*tmp)->left;
else if((*tmp)-data<data)
tmp=&(*tmp)->right;

else if(tmp->left&&tmp->right)

fine_max(&(*tmp)->left,&num);
(*tmp)->data=num;
}
else
{
(*tmp)=changr_mode(tree *root);
return ;
}
}
void *fine_max(tree **root,int *data)
{
tree **tmp=root;
while(*tmp)
{
if(tmp->right)
tmp=&(*tmp)->right;
else
*data=(*tmp)->data;
root=change_mode(root)
}
void *change_mode(tree *root)


{
tree *tmp=root;
while(root)
{
if(!root->left)
root=root->right;
else if(!root->left)
root=root->left;
free(tmp);
```

```
}
return root;
}



int main(void)
{
tree *root=NULL;
int data=[14]={50,45,73,32,48,16,37};
int i;
for(i=0;data[i];i++)
insert(&root,data[i]);
point(root);

Qpoint(&root);
delete(&root,45);

Return 0;

}
```