Computer Programming 1 Lab

2022-10-13

Outline

- Functions
- Recursive function
- Exercise4

Functions

Functions

- Pack codes together to perform specfic work.
- Return value based on the arguments given.

why use functions?

- Avoid rewriting same logic multiple times.

Functions

An easy example:

In math:

$$y = f(x) = 3x^2 + 5x + 4$$

In C:

```c
int f(int x){
    int ans = 3*x*x + 5*x + 4;
    return ans;
}
```

Call by value vs. Call by pointer:

```c
#include<stdio.h>

int f1(int x){
    x = 5*x + 4;
    return x;
}

int f2(int* x){
    *x = 5*(*x) + 4;
    return *x;
}

int main(){
    int x = 1;
    printf("x = %d\n", x);
    printf("f1(x) = %d\n", f1(x));
    printf("x = %d\n", x);
    printf("f2(&x) = %d\n", f2(&x));
    printf("x = %d\n", x);
    return 0;
}
```

Results

```
darkknive@nccucs108:~/codes/1111cp1/lab04$ gcc ./test.c
darkknive@nccucs108:~/codes/1111cp1/lab04$ ./a.out
x = 1
f1(x) = 9
x = 1
f2(&x) = 9
x = 9
```

Recursive function

Recursive function

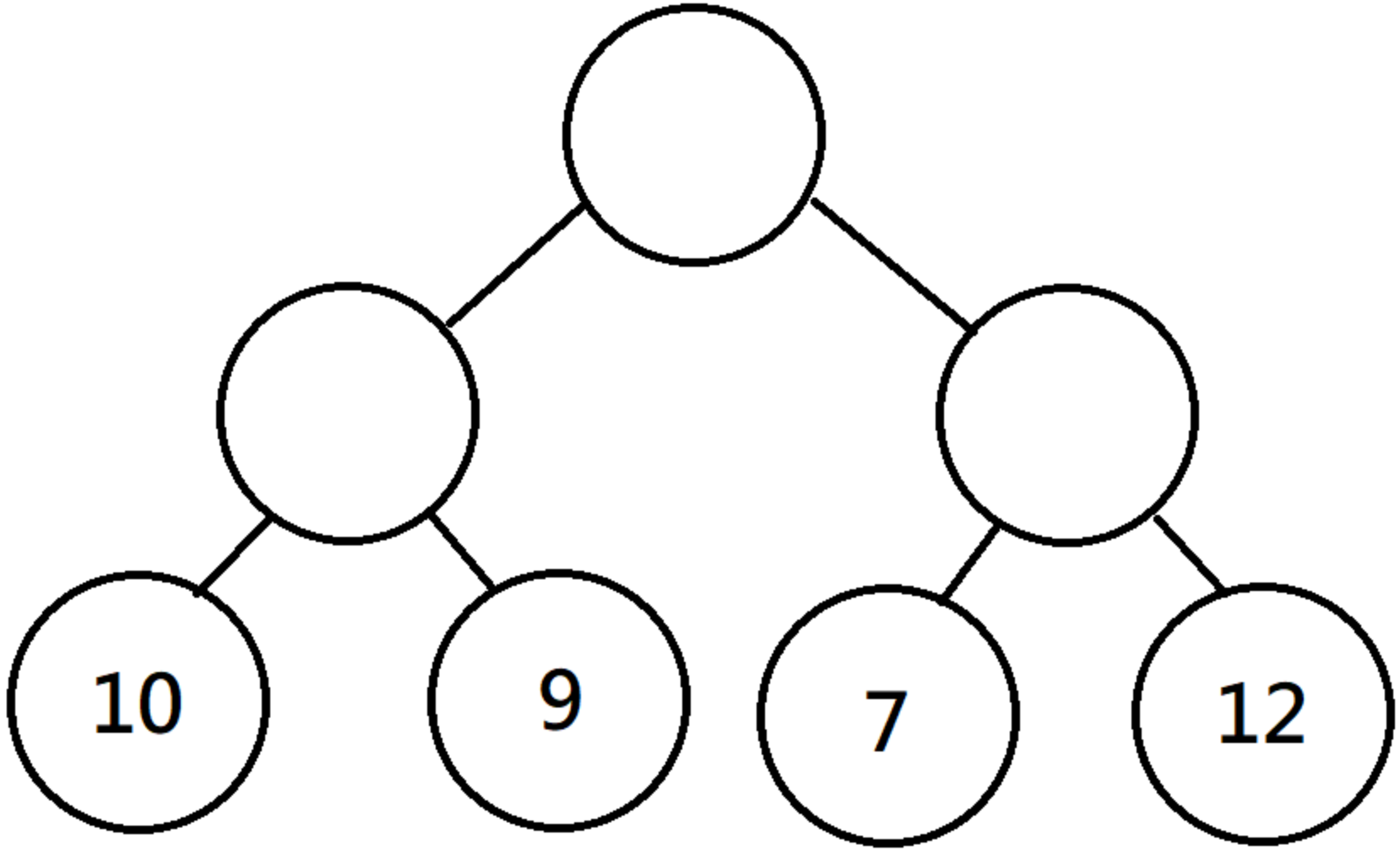
- Functions called by itself
- EX:
  - gcd (greatest common divisor)
  - DFS (depth first search)
  - Quicksort (divide and conquer)

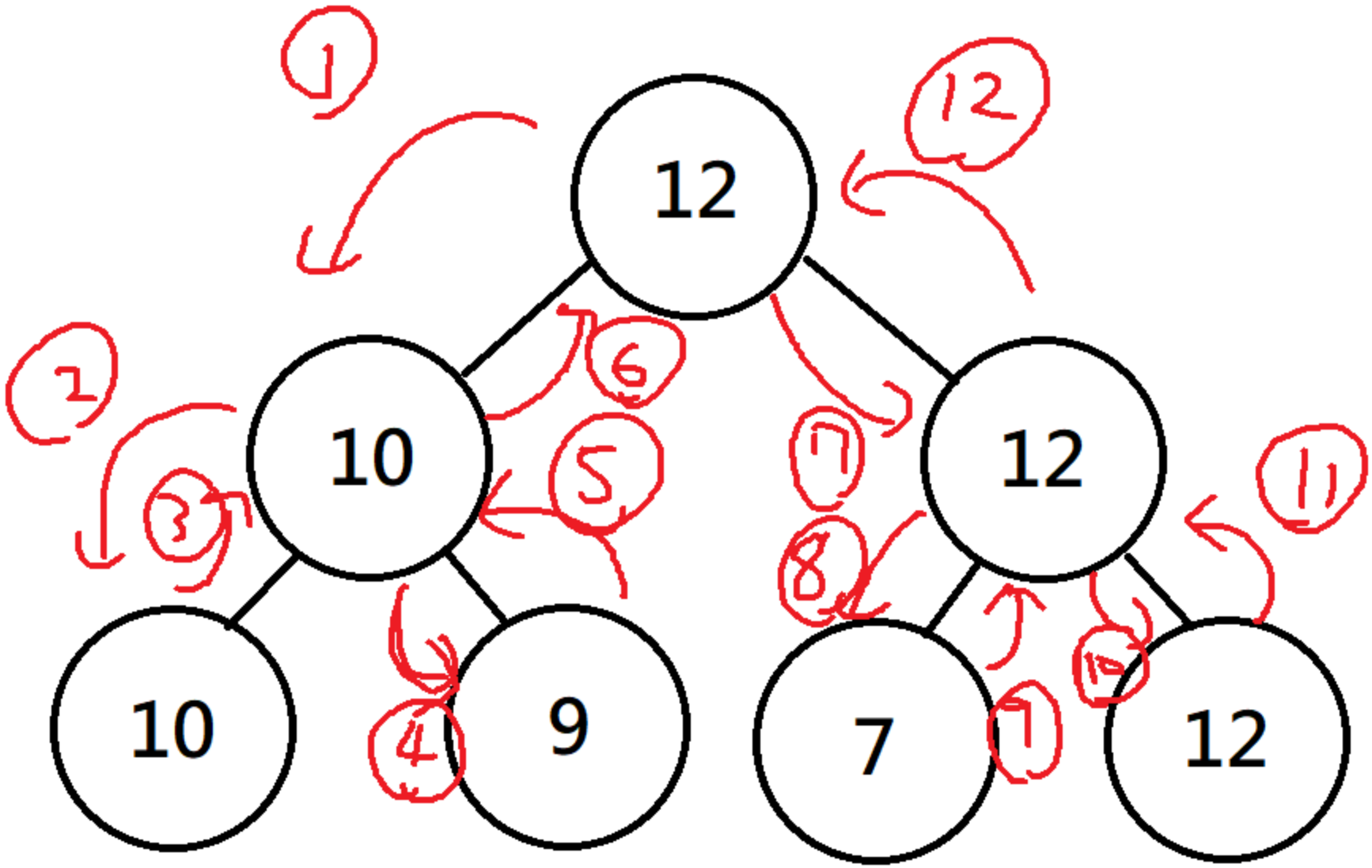
Recursive function

example DFS:

```
struct node;
typedef struct node Node;
struct node{
    int value = 0;
    Node* left = NULL;
    Node* right = NULL;
};

int DFS_max(Node* root){
    if (root->left == NULL && root->right == NULL){
        return root->value;
    }
    return max(DFS_max(root->left), DFS_max(root->right));
}
```


10
9
7
12


12
10
12
10
9
7
12

Exercise4

Any Question?

Course? Assignment? Exercise? TA?