CS 240 Programming in C

Recursion

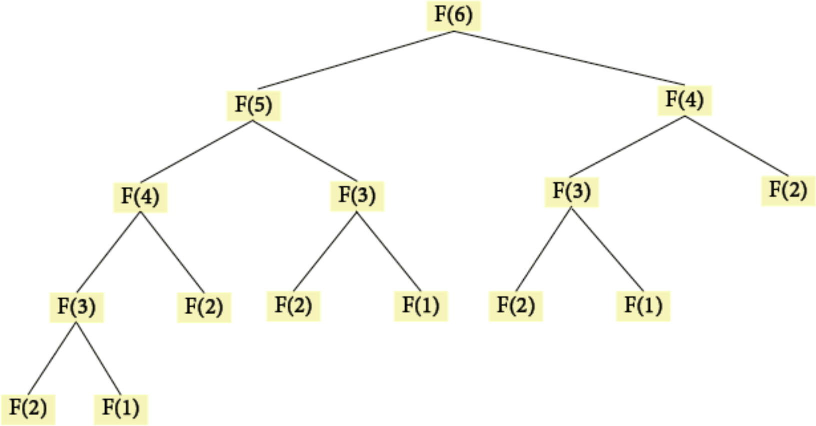
Oct 13, 2022

Recursion

Recursion is a method of solving a computational problem where the solution depends on solutions to smaller instances of the same problem. Recursion solves such recursive problems by using functions that call themselves from within their own code. The approach can be applied to many types of problems, and recursion is one of the central ideas of computer science.
[1]

[1] https://en.wikipedia.org/wiki/Recursion_(computer_science)

Recursion



Factorial using Iteration

```c
int factorial(int n)
{
    int fact = 1;
    for (int i = 1; i <= n; i++)
    {
        fact = fact * i;
    }
    return fact;
}

int main()
{
    int n = 6;
    printf("factorial(%d) = %d\n", n, factorial(n));
}
```

Factorial using Recursion

```
int factorial(int n)
{
    if (n == 0)
        return 1;
    else
        return n * factorial(n - 1);
}

int main()
{
    int n = 6;
    printf("factorial(%d) = %d\n", n, factorial(n));
}
```

How to read a recursive function

- Always identify the base case of the function before anything else.
- Pass arguments to the function that will immediately reach the base case.
- Identify the arguments that will at least execute the recursive function call once. [2]

[2] https://www.freecodecamp.org/news/what-is-recursion-in-javascript

How to write a recursive function

- Create a regular function with a base case that can be reached with its parameters
- Pass arguments into the function that immediately trigger the base case
- Pass the next arguments that trigger the recursive call just once.

Recursion

a recursive function might end up performing the same calculation with the same input multiple times. This means it could end up taking longer than the iterative alternative.

Memoization

A memoization function allows us to store input alongside the result of the calculation. Therefore, rather than having to do the same work again using the same input, it can simply return the value stored in the array/cache.

After Memoization

```
int fact(int n, int memo[])
{
    if (memo[n])
        return memo[n];
    else
        memo[n] = n * fact(n - 1, memo);
    return memo[n];
}
int main()
{
    int memo[] = {1, 0, 0, 0, 0, 0, 0, 0, 0, 0};
    int n = 5;
    printf("fact(%d) = %d\n", n, fact(n, memo));
    n = 6;
    printf("fact(%d) = %d\n", n, fact(n, memo));
}
```