Recursion


Canberra Institute
of Technology


aie
SPECIALIST EDUCATORS IN
GAMES, ANIMATION & FILM VFX

Contents

- What is recursion?
  - How to make a recursive function
  - How to end a recursive function
- Where is recursion useful?
- What are the problems with recursion?
  - Stack Overflows

Canberra Institute of Technology

aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

What is Recursion

- Recursion is when a function calls itself.

- The purpose of a recursive function:
  - To find the solution to a small piece of a bigger problem, thereby through iterations, solve the original (bigger) problem
  - It can be a useful programming technique

Canberra Institute of Technology

aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

How To Write A Recursive Function

- All you need to do to make a recursive function is to have the function call its own name.

- When the function calls itself, a new stack frame is pushed on the stack, and execution jumps back to the start of the function.

```
void RecursiveFunction()
{
    RecursiveFunction();
}
```

```
void InfiniteLoop()
{
    while(true)
    {

    }
}
```

ra Institute
chnology

aie
SPECIALIST EDUCATORS IN
GAMES, ANIMATION & FILM VFX

How To Write A Recursive Function

- If we look at the example to the right, if it ran it would
  - Enter the function
  - Call itself
  - Enter the function
  - Call itself
  - Enter the function
  - Call itself
  - Enter the function
  - Call itself
  - …

```
void RecursiveFunction()
{
    RecursiveFunction();
}
```

```
void InfiniteLoop()
{
    while(true)
    {

    }
}
```

ra Institute
chnology

aie
SPECIALIST EDUCATORS IN
GAMES, ANIMATION & FILM VFX

How to End a Recursive Function

- To end a recursive function, you need a branch somewhere that breaks the infinite recursion.

```
void RecursiveCount(int count)
{
    if ( count > 0 )
    {
        RecursiveCount(count - 1);
    }
    else
    {
        return;
    }
}
```

```
void CountLoop()
{
    int count = 10;
    while(count > 0)
    {
        count = count - 1;
    }
}
```

Canberra Institute of Technology

aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

Tail and NonTail Recursion

- A tail recursive function
  - Calls itself at the very end of the function, no statements follow and no recursive call before

```
void PrintNumberReverse(int x)
{
   if(x <= 0)
     return;

   cout << " " << x;
   PrintNumberReverse(x - 1);
}
```

```
int main(int argc, char * argv[])
{

   cout << "Recursion example " << endl << endl;

   PrintNumberReverse(10);

   return 0;
}
```

Canberra Institute of Technology

aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

NonTail Recursion

- Example

```
void nonTail(int i)
{
   if(i > 0)
   {
      nonTail(i - 1);
      cout << i << ' ';
      nonTail(i - 1);
   }
}
```

- What will this print out if its argument is set to 3?
  - the output will be 1, 2, 1, 3, 1, 2, 1

Canberra Institute of Technology
aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

More examples

- A factorial of n is the product of all positive integers less than or equal to ‘n’. Recursion can be used to find the factorial of ‘n’

```
// Factorial: compute n!
int Factorial(int n)
{
   if(n <= 1)
     return 1;

    return n*(Factorial(n-1));
}
```

Canberra Institute of Technology

aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

Recursion Calls

- What happens if we call Factorial(3)?
- Calculate Factorial(3) from the last slide?
  - Call 1: Factorial(3)
  - Call 2: Factorial(2)
  - Call 3: Factorial(1)
  - Call 4: Factorial(0)
  - Return: Factorial(0)
  - Return: Factorial(1)
  - Return: Factorial(2)
  - Return: Factorial(3)

Canberra Institute of Technology

aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

Recursion Limitations

- Because recursion returns to the state that calls the function, if recursive function do not have an explicit early exit condition, can cause the stack to overflow

- There is a limit to the number of times a function can recurse
  - Depends on the stack size (default is 1MB), and the complexity and size of the recursive function

- You need to careful you do not cause a stack overflow.

Canberra Institute of Technology

aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

Summary

- Recursion is when a function calls itself
- Recursion solves problems that can be broken up into smaller versions of themselves.
- Used mainly for trees and graphs.

Canberra Institute of Technology
aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

References


Canberra Institute
of Technology


aie
SPECIALIST EDUCATORS IN
GAMES, ANIMATION & FILM VFX