

Laboratório: Recursividade
Disciplina: FGA0147 - Estruturas de Dados e Algoritmos
Prof. Dr. Nilton Correia da Silva
Faculdade UnB Gama - FGA
Universidade de Brasília - UnB

Aplicação de Técnica de Programação por Recursividade na Função de Ackermann.

Referência: Ackermann function

$$A(x, y) \equiv \begin{cases} y + 1 & \text{if } x = 0 \\ A(x - 1, 1) & \text{if } y = 0 \\ A(x - 1, A(x, y - 1)) & \text{otherwise.} \end{cases}$$

Gama, Distrito Federal

Sumário

1	Introdução	3
2	Função de Ackermann	4
3	Objetivo	5
3.1	Requisitos	5
3.2	Exemplo de Entrada e Saída	5
4	Dicas	5



Orientações

Laboratório da disciplina de Estruturas de Dados e Algoritmos. Este laboratório foi elaborado considerando que sua resolução deve ser realizada de forma individual.

A solução deverá ser em linguagem C ou C++. No caso do acadêmico adotar C++, poderá se valer de uma solução orientada a objetos, contudo sua solução não deve usar classes e contêineres já prontos da linguagem C++ - por exemplo, algoritmos de ordenação, busca, objetos vector, list, matrix, etc.



1 Introdução

A técnica de programação por recursividade é uma abordagem que permite que uma função chame a si mesma, com diferentes parâmetros, para resolver um problema. A recursividade é uma técnica muito poderosa e flexível, que pode ser utilizada para resolver uma grande variedade de problemas em programação.

A recursividade é baseada em um conceito matemático chamado de indução, que consiste em provar uma proposição para um conjunto infinito de casos a partir de alguns casos base e de uma regra de indução. Na programação, os casos base correspondem a situações simples em que é possível resolver o problema diretamente, sem precisar fazer mais chamadas recursivas. A regra de indução é uma relação que descreve como o problema pode ser resolvido a partir de problemas menores, que são resolvidos pela chamada recursiva da função.

A recursividade tem várias vantagens, como a elegância da solução, a simplicidade do código e a facilidade de compreensão e manutenção do código. Porém, a recursividade também tem algumas desvantagens, como a possibilidade de gerar um grande número de chamadas recursivas e o risco de ocorrer um estouro de pilha (stack overflow) em casos de recursividade excessiva.

Por isso, é importante escolher cuidadosamente os casos base e a regra de indução, para garantir que a recursividade seja bem definida e não gere problemas de desempenho ou estouro de pilha. Além disso, é possível transformar uma solução recursiva em uma solução iterativa, o que pode ser mais eficiente e seguro em alguns casos. A escolha entre soluções recursivas e iterativas deve ser feita de acordo com as necessidades específicas de cada problema.



2 Função de Ackermann

A função de Ackermann é uma função matemática que foi proposta pelo matemático alemão Wilhelm Ackermann em 1928. Ela é um exemplo clássico de uma função recursiva que cresce muito rapidamente e é usada para avaliar a eficiência de algoritmos recursivos.

A função de Ackermann é definida recursivamente por meio de três casos:

- Se $m = 0$, então $\text{ackermann}(m, n) = n + 1$.
- Se $n = 0$ e $m > 0$, então $\text{ackermann}(m, n) = \text{ackermann}(m - 1, 1)$.
- Se $m > 0$ e $n > 0$, então $\text{ackermann}(m, n) = \text{ackermann}(m - 1, \text{ackermann}(m, n - 1))$.

Essa definição pode ser um pouco difícil de entender à primeira vista, mas a ideia básica é que a função de Ackermann usa a recursividade para definir uma série de operações matemáticas cada vez mais complexas.

O primeiro caso é o mais simples, onde a função simplesmente retorna $n + 1$ se m é igual a zero. O segundo caso é um pouco mais complicado, onde a função chama a si mesma com $m - 1$ e 1 como parâmetros, e o resultado dessa chamada é retornado. Finalmente, o terceiro caso é o mais complexo, onde a função chama a si mesma duas vezes com parâmetros diferentes antes de retornar o resultado.

O valor da função de Ackermann cresce muito rapidamente conforme aumentamos os valores de m e n , o que torna a função útil para testar a eficiência de algoritmos recursivos. Na verdade, a função cresce tão rapidamente que é possível que ela não possa ser calculada em uma máquina convencional para valores de m e n suficientemente grandes.

Apesar de sua aparência simples, a função de Ackermann é uma das funções recursivas mais estudadas e desafiadoras da matemática, e continua sendo objeto de estudo ativo até hoje.



3 Objetivo

Você deve implementar um programa em linguagem C ou C++ que receba dois números inteiros positivos **m** e **n** como entradas e calcule o valor da função de Ackermann usando recursão.

3.1 Requisitos

1. Deve utilizar recursão para calcular o valor da função de Ackermann. Crie uma função chamada **AckermannRecursivo**.
2. A função main deve ler dois números inteiros positivos **m** e **n** do usuário, chamar a função **AckermannRecursivo** para calcular o valor da função de Ackermann e imprimir o resultado na tela.
3. Deve validar a entrada do usuário para garantir que **m** e **n** sejam números inteiros positivos.

3.2 Exemplo de Entrada e Saída

Digite dois números inteiros positivos: 3 4

O valor da função de Ackermann para m=3 e n=4 é 125

4 Dicas

- Você pode implementar a função **AckermannRecursivo** da seguinte maneira: se m for igual a 0, retorne $n + 1$. Se m for maior do que 0 e n for igual a 0, chame recursivamente a função de Ackermann com os parâmetros m - 1 e 1. Se m e n forem maiores do que 0, chame recursivamente a função de Ackermann com os parâmetros m - 1 e o resultado da chamada recursiva com os parâmetros m e n - 1.
- Na função main, você pode utilizar a função scanf para ler a entrada do usuário e a função printf para imprimir o resultado na tela.

