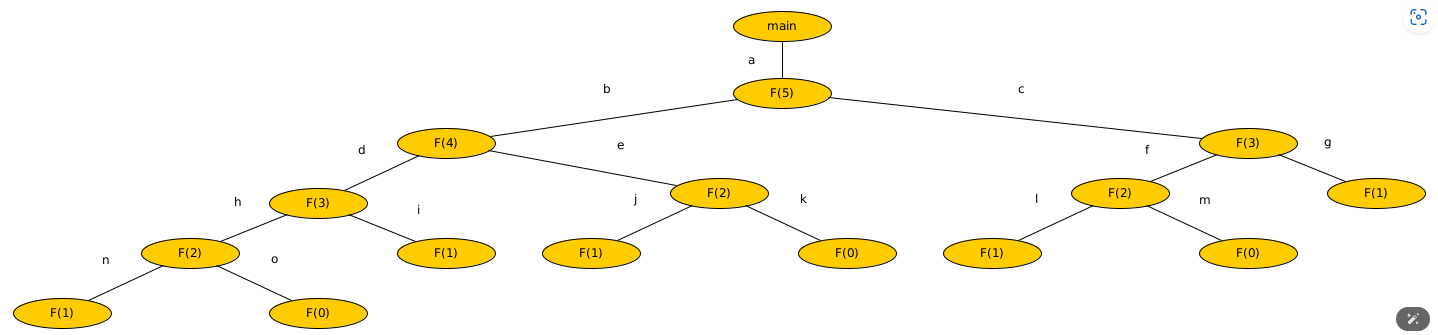
É comum usar árvores para esquematizar a explicação da execução de funções recursivas. Na árvore abaixo instâncias de execução da função são representadas por nós e chamadas/retornos, às vezes implicitamente, são representadas por arestas.



Essa específica árvore representa o cálculo de F(5), o quinto elemento da sequência de Fibonacci.  
  
Neste caso, F(n)=F(n−1)+F(n−2)F(n)=F(n−1)+F(n−2) onde F(0)=1F(0)=1 e F(1)=1F(1)=1  
  
O cálculo foi implementado em `Fiborec.c`.  
  
Uma explicação sucinta (e truncada) pode ser: *Deseja-se calcular F(5). Calcular F(5) requer, antes, calcular F(4) e F(3), calcular F(4) requer calcular F(3) e F(2), ... enumerando todos os nós*.  
  
Em explicações (por exemplo em aula expositiva) do funcionamento do algoritmo, até certo ponto, abstraem-se ordens de chamada e retorno da (instância da) função, algo que só se torna claro quando a função é implementada, executada e estudada.  
  
1. A ordem temporal de chamadas e de retornos pode ser obtida observando a ordem de impressão das mensagens na tela  
2. A relação entre a árvore da figura e a árvore obtida pela execução do programa pode ser *não intuitiva*.pois os retornos das funções são omitidos (portanto uma aresta é percorrida duas vezes, uma na chamada ("descida"), outra no retorno ("subida")) e as chamadas são feitas "em profundidade" e não "em largura"  
  
A cada linha impressa na tela, uma aresta foi percorrida (visitada). A ordem de visitação das arestas durante a execução de fiborec é:

| linha | aresta |

| ----- | ------ |

| 1 | a |

| 2 | b |

| 3 | d |

| 4 | h |

| 5 | n |

| 6 | n |

| 7 | o |

| 8 | o |

| 9 | h |

| 10 | i |

| 11 | i |

| 12 | d |

| 13 | e |

| 14 | j |

| 15 | j |

| 16 | k |

| 17 | k |

| 18 | e |

| 19 | b |

| 20 | c |

| 21 | f |

| 22 | l |

| 23 | l |

| 24 | m |

| 25 | m |

| 26 | f |

| 27 | g |

| 28 | g |

| 29 | c |

Note que, da maneira como o programa foi escrito, a aresta 'a' tem somente uma mensagem associada (pois não é escrita na tela a mensagem correspondente ao retorno da chamada).  
  
**O Exercício consiste em fazer o mesmo para o programa hanoi.c.**  
ié:  
  
1. ajustar o programa hanoi.c para apresentar mensagens a cada chamada e retorno;  
2. desenhar a árvore de execução;  
3. associar a cada aresta da árvore a chamada e retorno correspondentes.  
  
A árvore acima foi desenhada usando o editor de gráficos yEd (https://www.yworks.com/products/yed). O arquivo que corresponde à árvore é fibo.graphml.  
  
Envio das respostas:  
Questão 1: anexar o código de hanoi.c ajustado por você para apresentar as mensagens;  
Questão 2: anexar a figura  (png) gerado pela exportação da árvore que você desenhou no yEd;  
Questão 3: apresentar a tabela com a visitação das arestas usando o editor do e-disciplinas.  
  
Data de entrega: 20 de outubro, 23h.

As arestas em branco na tabelas ocorre devido ao uso do printf para representar quando ocorre a mudança, portanto não há utilização de aresta para chamada(subida) ou retorno(retorno)