UNIVERSIDADE FEDERAL DE UBERLÂNDIA - Campus Monte Carmelo

Faculdade de Computação

Bacharelado em Sistemas de Informação

Disciplina: Estruturas de Dados 2

Professor: Dr. Carlos C. M. Tuma

2º Trabalho de Estruturas de Dados 2

Marcus Antonio Rufino Ribeiro 31511BSI023

MONTE CARMELO

2019

SUMÁRIO

1. INTRODUÇÃO

Problema das 8 rainhas

O objetivo desse trabalho foi criar um programa para calcular as possíveis possibilidades de colocar 8(oito) rainhas em um tabuleiro de 8x8 sem que nenhuma rainha ameace uma rainha que já foi posta no tabuleiro. O programa foi feito usando recursão e um vetor de 8 posições para salvar as posições das rainhas.

Possibilidade 1

TABULEIRO

	0	1	2	3	4	5	6	7
0	R	.	.	.	.	.	.	.
1	.	.	.	.	.	.	R	.
2	.	.	.	.	R	.	.	.
3	.	.	.	.	.	.	.	R
4	.	R	.	.	.	.	.	.
5	.	.	.	R	.	.	.	.
6	.	.	.	.	.	R	.	.
7	.	.	R	.	.	.	.	.

coluna [0] = linha 0
coluna [1] = linha 4
coluna [2] = linha 7
coluna [3] = linha 5
coluna [4] = linha 2
coluna [5] = linha 6
coluna [6] = linha 1
coluna [7] = linha 3

2. DESCRIÇÃO DA SOLUÇÃO

O programa não possui segredos não precisa de nenhum dado de entrada, e só compilar e executar.apos a execução do programa ele retorna para cada possibilidade as posições que as rainhas 8 rainhas tem que ser colocadas no tabuleiro, no final da execução do programa as possibilidades são gravadas em um arquivo .txt

3. FUNCIONALIDADES

Função rainhas função feita para verificar se e possível colocar uma rainha naquela posição ela retorna um valor inteiro, se o retorno for 1 não e possível adicionar uma a rainha naquela posição, se o returno for 0 e possível adicionar a rainha.

```
int  rainhas(int c)
{
   int a=0,i;
   for (i=0; i<c; i++)
   {
      if (r[i]==r[c]) //verifica a linha
      {
         return 1;
      }
      if (r[c]==(r[i]+(c-i))) // verifica a diagonal principal
      {
         return 1;
      }
      if (r[c]==(r[i]-(c-i))) //verifica a diagonal secundaria
      {
         return 1;
      }
   }
   return 0;
}
```

Função busca rainha essa função serve para verificar as possiçoes do tabuleiro, passando um inteiro como parâmetro começando de 0 eu vejo se esse valor for =-1(igual a menos um) não tenho mais posição nesse tabuleiro, se meu valor for = 8(igual a oito) achou uma resposta aceitável, chamo a função imprimir decremento o meu valor e busco mais resultados,se minha rainha na possicao do inteiro for maior que 7 essa posição recebe 0 volto uma rainha e tento colocar na próxima casa chamando a recursão.

```
Int buscaRainha(int c)
{
   while(c!=-1){
   if(c==8)  // limite do tabuleiro
   {
      imprimir(c);// printa o tabuleiro e as posições
      c--;
```

```
        r[c]++;
        buscaRainha(c);
        return 1;
    }
    if(c==-1)  // fim das possibilidades
    {
        printf("Fim das posibilidades\n");
        return 1;

    }
    if(r[c] > 7)       {
        r[c]=0;
        if((c-1)>-1)
            r[c-1]++;
        buscaRainha(rainhas(c-1));
        return 1;
    }
    if(rainhas(c)==0)
    {
        c++;
        buscaRainha(c);
        return 1;
    }
    else
    {

        r[c]++;

        buscaRainha(c);
        return 1;
        }
    }
    return 0;
}
```

Função imprimir essa função e soh pra imprimir o resultado e salvar o mesmo no arquivo.toda vez que eu chamo essa função minha variável possibilidade e incrementada em +1(mais um).

```
void imprimir(int c)
{
    int i;
    fprintf(arq,"Possibilidade %d\n\t\t\tTABULEIRO\n\n",posibilidades);
    printf("Possibilidade %d\n\t\t\tTABULEIRO\n\n",posibilidades);
```

```
    for( i=0; i<8; i++)
    {
        fprintf(arq,"\t%d",i);//gravando no arquivo o inteiro,representando a
coluna
        printf("\t%d",i);
    }
    for( i=0; i<=c-1; i++)
    {
        fprintf(arq,"\n\n%d ",i);//
        printf("\n\n%d-> ",i);
        int j;
        for(j=0; j<=c-1; j++)
        {
            if(r[j]==i)
            {
                fprintf(arq,"%s","\tR");
                printf("\tR");
            }
            else
            {
                fprintf(arq,"%s","\t.");
                printf("\t.");
            }
        }
    }
    printf("\n\nPosicoes das rainhas:\n");
    for(i=0; i<8; i++)
    {
        printf("coluna [%d] = linha %d\n",i,r[i]);
        fprintf(arq,"\ncoluna [%d] = linha %d\n",i,r[i]);
    }
    posibilidades++;
}
```

4. LIMITAÇÕES

Como o programa não possui dados de entrada ele também não possui limitações, o único problema foi que eu achei foi que eu queria achar todas as possibilidades e quando eu compilo o programa me mostras 250 possibilidades e a memória do meu computador estoura.