DOCUMENTAÇÃO TRABALHO PRÁTICO 1
Vinícius de Castro Mendes Morais - 2018020867

Fase de Especificação

O projeto foi feito com uma técnica que minimiza o número de passos, sendo utilizado sqrt(n) passos. Usou-se interações a fim de contabilizar as movimentações do “para cima”, “para baixo”, “para esquerda” e “para direita”. Um padrão foi estabelecido e será explicado na fase projeto.

Projeto

A fim de procurar um padrão que relaciona as coordenadas do ponto com o número ordinal da espiral quadrática em que ele é representado, foi feito o seguinte processo.

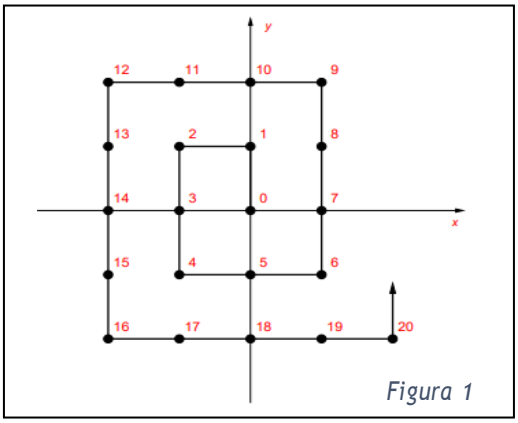

Figura 1

De início, foi inserido um “while” de modo que é possí- vel realizar todos os testes sem a necessidade de ficar executando o programa diversas vezes. Para encerrá- lo, basta inserir um valor de n<0.

Então, será feito o cálculo da raiz quadrada inteira do número do ponto em que se deseja saber a coorde- nada e, por meio do resto da divisão, descobrir se essa raiz é par, o que torna possível saber qual a posição dele em relação aos pontos que estão na mesma diagonal do 16 e 4 e na diagonal do 1 e 9, seguindo a figura 1. Exemplo: ponto 10. Sua raiz quadrada é aproximadamente 3,1623, portanto, sua raiz quadrada in- teira é 3. Ao dividir por 2, tem-se resto 1, mostrando que é uma raiz ímpar. O código dessa operação está representado na figura 2. Como sua raiz é ímpar, a sua posição é entre uma raiz impar inteira e uma raiz par inteira, sendo a par maior e a impar a menor, (raiz impar inteira < raiz ímpar < raiz par inteira). Caso a raiz fosse par, aconteceria o contrário (raiz par inteira < raiz par < raiz impar inteira). Assim, os únicos números que possuem raiz natural são os membros das diagonais citadas acima. Ou seja, é possível prever entre quais pontos da diagonal o ponto inserido está localizado. O valor de t será usado para a contagem dos movimentos já feitos, explicado mais à frente. Seguindo o raciocínio, o ponto inicial será algum desses pontos da diagonal e deve-se prever qual será o movimento feito para atingir o número seguinte. No caso de raízes inteiras ímpares, os pontos de inicio serão os que possuem o eixo das ordenadas e das abcissas (x, y) positivas, enquanto os de raiz inteira par, negativas. Para saber as coordenadas do ponto de início, no caso das raízes impares, basta dividir por 2 a raiz. Esse valor representará o x, e o y será a subtração da raiz por x. (Figura 3). Já para os de raiz inteira ímpar, divide-se a raiz por 2 e multiplica por -1 -- devido a sua posição em quadrante negativo -- para achar x, que é igual ao y (Figura 4). Seguindo o exemplo com o ponto 10, a parte inteira de sua raiz é 3 e, portanto, está entre os pontos 9 (raiz = 3) e 16 (raiz = 4). Ao dividir 3 por 2, a parte inteira será 1, ou seja, x inicial = 1. Y=Raiz – X, logo, y inicial = 2.

```
scanf("%d", &n);
h=sqrt(n);
t=h*h;
if (h%2==1)
```
Figura 2

```
if (h%2==1)
{
    x=h/2;
    y=h-x;
```

Figura 3

```
else
{
    x=(h/2)*(-1);
    y=x;
```

Figura 4

Então deve-se iniciar os movimentos. O valor inteiro da raiz é o mesmo que o número de movimentos máximo que se consegue fazer no eixo x. Assim, foi uti- lizado o comando *for* que finaliza ao atingir o valor de h ou então o número que representa o ponto. Será sempre acrescido 1 ao valor de t após um movimento, de forma a contabilizar quantos já foram feitos. (Figuras 5 e 6).

```
for (i=0; i<h; i++)
{

    if (n==(h*h)+i)
        break;
    t++;
    x++;
```

Figura 5 – para raiz inteira par

```
for (i=0; i<h; i++)
{

    if (n==(h*h)+i)
        break;
    t++;
    x--;
```

Figura 6 – para raiz inteira ímpar

Já no eixo y, os movimentos são determinados pela quantidade que falta até chegar no número desejado. (Figuras 7 e 8).

```
for (j=t; j<n; j++)
{
    if (n==(h*h))
        break;

    y--;

}
```

Figura 7

```
for (j=t; j<n; j++)
{
    if (n==(h*h))
        break;

    y++;
```

Figura 8

Continuando o exemplo com o ponto 10, então o máximo de movimentos que pode ser feito no eixo x é 3 e para a esquerda. Porém, ao ser feito um movimento já atinge o número do ponto. Será contabilizado -1 no valor de x, já que foi feito um movimento para a esquerda e o comando for é finalizado. Será impresso na tela as coordenadas do ponto. (Figura 9).

```
"C:\Users\vcm11\Documents\Estudos\2019 1\AEDS 2\Espiral\main.exe"
10
Coordenada do ponto 10: (0,2)
-1

Process returned 0 (0x0)   execution time : 9.774 s
Press any key to continue.
```

Figura 9

Assim, após todos os movimentos, será atingido o número do ponto e, portanto, será determinada a coordenada do ponto. Então, é pedido para se inserir outro número, para prosseguir os testes. Para encerrar o programa, basta inserir um número inteiro menor

que 0.

Implementação
Os comandos utilizados com a intenção de repetição foram *for* e *while*, já o condicional foi o *if.*

Comando *for*: usado para a repetição dos movimentos até serem atingidos os valores necessários.

Comando *while*: usado para manter o programa em execução para ser possível fazer diversos testes sem a necessidade de inicializá-lo novamente. Encerra ao inserir n<0.

Comando *if*: usado com o intuito de monitorar se já foi atingido o valor desejado, a fim de encerrar o *for*.