

Coloração

(IED-001)

Prof. Dr. Silvio do Lago Pereira

Departamento de Tecnologia da Informação

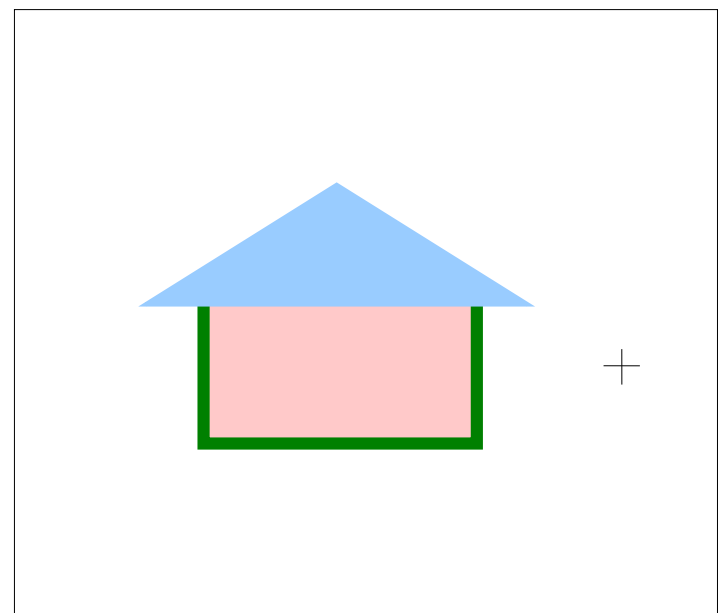
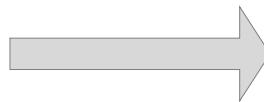
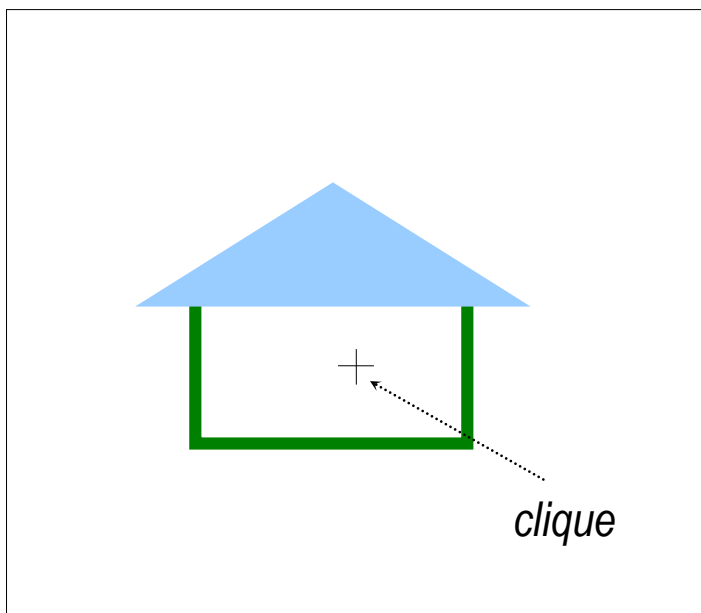
Faculdade de Tecnologia de São Paulo



Coloração

Coloração

é uma operação usada para colorir uma região de uma imagem composta de pixels.



colorir



nova cor



colorir



nova cor

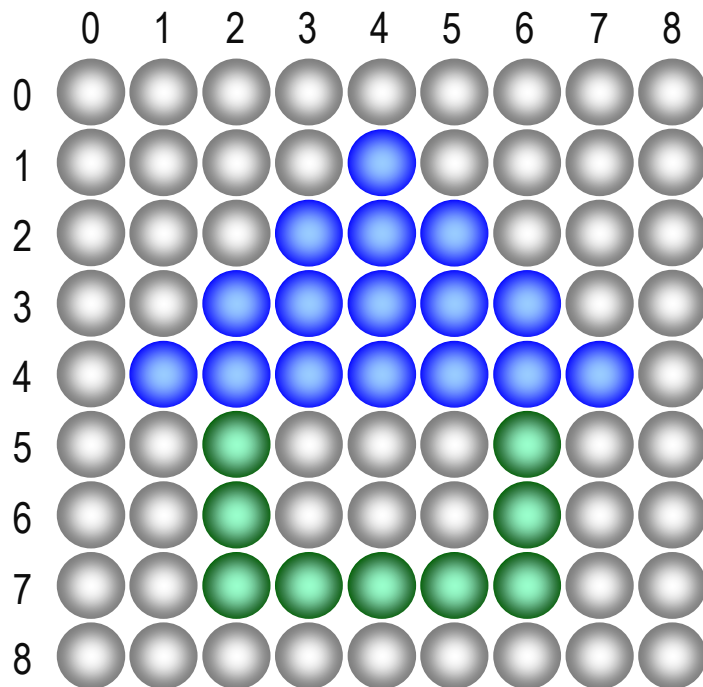
Como representar uma imagem e implementar a operação de coloração de regiões?



Coloração

Imagem

é uma matriz bidimensional de pixels, representados na memória por números naturais.



(a) imagem no vídeo

	0	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0
2	0	0	0	1	1	1	0	0	0
3	0	0	1	1	1	1	1	0	0
4	0	1	1	1	1	1	1	1	0
5	0	0	2	0	0	0	2	0	0
6	0	0	2	0	0	0	2	0	0
7	0	0	2	2	2	2	2	0	0
8	0	0	0	0	0	0	0	0	0

(b) representação na memória

As quantidades máximas de cores e pixels numa imagem dependem da capacidade do vídeo!



Coloração

Cores

no Pelles C são representadas em um sistema RGB que usa apenas 4 bits por cor.

Cor	Número	RGB
preto	0	0000
azul	1	0001
verde	2	0010
ciano	3	0011
vermelho	4	0100
magenta	5	0101
amarelo	6	0110
cinza claro	7	0111

Cor	Número	RGB
cinza escuro	8	1000
azul claro	9	1001
verde claro	10	1010
ciano claro	11	1011
vermelho claro	12	1100
magenta claro	13	1101
amarelo claro	14	1110
branco	15	1111

A função `_textcolor()`, usada para seleção de cor, está declarada em `conio.h`!

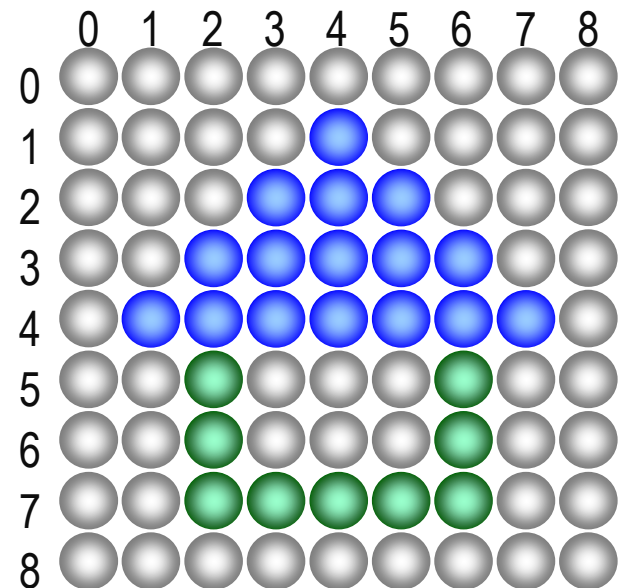


Coloração

Exemplo 1. Representação da imagem

```
#define dim 9
```

```
int I[dim][dim] = {  
    {0, 0, 0, 0, 0, 0, 0, 0, 0},  
    {0, 0, 0, 0, 1, 0, 0, 0, 0},  
    {0, 0, 0, 1, 1, 1, 0, 0, 0},  
    {0, 0, 1, 1, 1, 1, 1, 0, 0},  
    {0, 1, 1, 1, 1, 1, 1, 1, 0},  
    {0, 0, 2, 0, 0, 0, 2, 0, 0},  
    {0, 0, 2, 0, 0, 0, 2, 0, 0},  
    {0, 0, 2, 2, 2, 2, 2, 0, 0},  
    {0, 0, 0, 0, 0, 0, 0, 0, 0}  
};
```



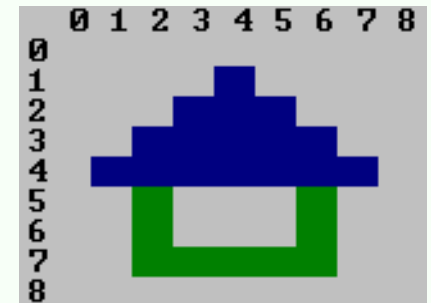
Para simplificar, usaremos uma matriz quadrada de inteiros para representar a imagem!



Coloração

Exemplo 2. Exibição da imagem

```
void exiba(int I[dim][dim]) {  
    for(int i=-1; i<dim; i++) {  
        _textcolor(8);  
        for(int j=-1; j<dim; j++)  
            if( i<0 && j<0 ) printf("  ");  
            else if( i<0 ) printf("%2d",j);  
            else if( j<0 ) printf("\n%2d",i);  
            else {  
                _textcolor(I[i][j]);  
                printf("%c%c",219,219);  
            }  
        }  
    }  
    _textcolor(8);  
}
```





Coloração

Exercício 1. Programa para representação e exibição da imagem

Complete e teste o programa a seguir.

```
#include <stdio.h>
#include <conio.h>

#define dim 9

int I[dim][dim] = {
    ...
};

void exiba(int I[dim][dim]) {
    ...
}

int main(void) {
    exiba(I);
    puts("\n");
    return 0;
}
```



Coloração

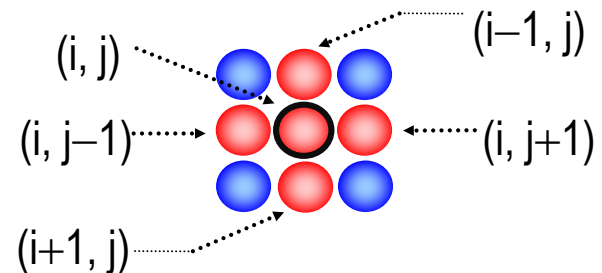
Coloração limitada por área

colore apenas pontos vizinhos que têm a mesma cor do ponto inicial da coloração.

Algoritmo de coloração:

Dados um pixel p e uma nova cor n , a coloração da região de p é feita do seguinte modo:

- Crie uma fila vazia F .
- Obtenha a cor atual a de p .
- Mude a cor de p para n .
- Insira p na fila F .
- Enquanto a fila F não estiver vazia, faça :
 - Remova um ponto p de F .
 - Para cada vizinho q de p , que tenha a cor a , faça:
 - Mude a cor de q para n .
 - Insira q na fila F .

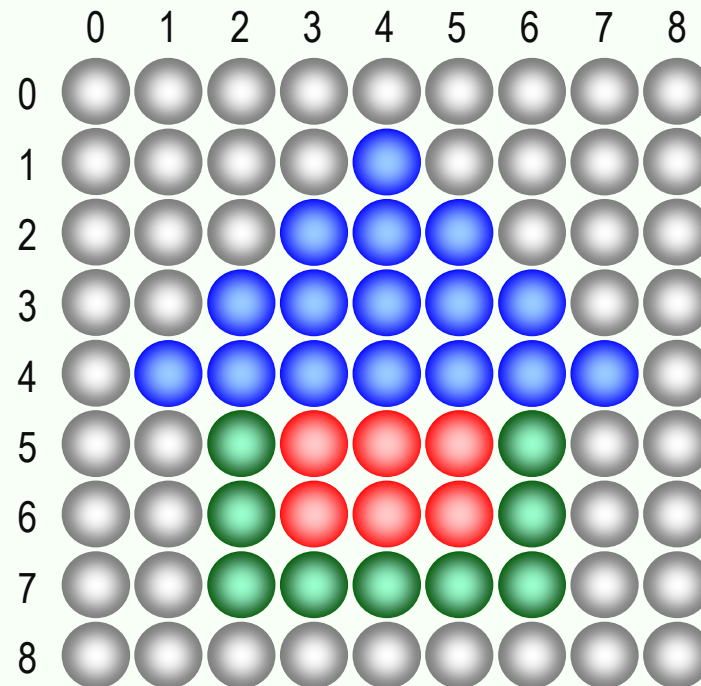




Coloração

Exemplo 3. Funcionamento do algoritmo de coloração

$p = (5, 4)$
 $n = 4$ (vermelho)
 $a = 0$ (preto)



(b) $F = []$

A fila mantém pixels que já foram coloridos, mas cujos vizinhos ainda não foram inspecionados!



Coloração

Exemplo 4. Tratamento de bordas e coordenadas

```
#define cor(i,j) (i>=0 && i<dim && j>=0 && j<dim ? I[i][j] : -1)
#define par(i,j) ((i)*100+(j))
#define lin(p) ((p)/100)
#define col(p) ((p)%100)
```

	0	1	2	3	4	5	6	7	8	
	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
0	-1	0	0	0	0	0	0	0	0	-1
1	-1	0	0	0	0	1	0	0	0	-1
2	-1	0	0	0	1	1	1	0	0	-1
3	-1	0	0	1	1	1	1	1	0	-1
4	-1	0	1	1	1	1	1	1	0	-1
5	-1	0	0	2	0	0	0	2	0	-1
6	-1	0	0	2	0	0	0	2	0	-1
7	-1	0	0	2	2	2	2	2	0	-1
8	-1	0	0	0	0	0	0	0	0	-1
	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1



Coloração

Exemplo 5. Coloração da imagem

```
void colorir(int I[dim][dim], int i, int j, int n) {
    Fila F = fila(dim*dim);
    int a = I[i][j];
    I[i][j] = n;
    enqueue(par(i,j),F);
    while( !vaziaf(F) ) {
        int p = dequeue(F);
        i = lin(p);
        j = col(p);
        if( cor(i-1,j)==a ) { I[i-1][j]=n; enqueue(par(i-1,j),F); }
        if( cor(i,j+1)==a ) { I[i][j+1]=n; enqueue(par(i,j+1),F); }
        if( cor(i+1,j)==a ) { I[i+1][j]=n; enqueue(par(i+1,j),F); }
        if( cor(i,j-1)==a ) { I[i][j-1]=n; enqueue(par(i,j-1),F); }
    }
    destroy(&F);
}
```



Coloração

Exemplo 6. Programa principal

```
#include <stdio.h>
#include <conio.h>
#include "fila.h"
...
int main(void) {
    int i, j, n;
    while( 1 ) {
        exiba(I);
        printf("\n\nNova cor (ou -1 para sair)? ");
        scanf("%d",&n);
        if( n<0 ) break;
        do {
            printf("Posicao? ");
            scanf("%d,%d",&i,&j);
        } while( i<0 || i>=dim || j<0 || j>=dim );
        colorir(I,i,j,n);
    }
    return 0;
}
```



Coloração

Exercício 1. Correção do programa

Execute o programa do Exemplo 6 e veja o que acontece quando o usuário tenta preencher uma região com a mesma cor que ela já tem. Altere a função `colorir()` para corrigir o problema.

Exercício 2. Entrada de arquivo

A função a seguir inicia uma matriz `I` com dados lidos de um arquivo. Por exemplo, para iniciar uma matriz com `imagem.txt`, basta chamar `inicia(I, "imagem.txt")`. Usando essa função, altere o programa de coloração anterior para que ele leia a imagem de um arquivo.

```
void inicia(int I[dim][dim], char *s) {  
    FILE *a = fopen(s, "r");  
    if( !a ) {  
        puts("arquivo não encontrado");  
        abort();  
    }  
    for(int i=0; i<dim; i++)  
        for(int j=0; j<dim; j++)  
            fscanf(a, "%d", &I[i][j]);  
    fclose(a);  
}
```

imagem.txt

2	2	2	2	6	2	2	2	2
2	2	2	6	6	6	2	2	2
2	2	6	6	1	6	6	2	2
2	6	6	1	1	1	6	6	2
6	6	1	1	7	1	1	6	6
2	6	6	1	1	1	6	6	2
2	2	6	6	1	6	6	2	2
2	2	2	6	6	6	2	2	2
2	2	2	2	6	2	2	2	2

Fim

