

Departamento de Informática – UFPR

Algoritmos e Estruturas de Dados I

Prof.^ª Dr.^ª Mariane Cassenote

mariane@inf.ufpr.br

Refinamentos Sucessivos

Campo Minado

- Conteúdo
 - Registros
 - Matrizes
 - TADs
 - Pilha
 - Sentinelas
 - Modularidade
 - Refinamentos sucessivos

Refinamentos sucessivos

- Começa pelo programa mais abstrato possível
- A cada etapa, detalha uma peça
- Funções e procedimentos escondem a complexidade de cada nível
- O programa principal conta a história. Os detalhes ficam embaixo

Programa principal

```
begin
  inicializar_campo(c);
  imprimir_campo(c);
  repeat
    ler_jogada(x, y, c);
    executar_jogada(x, y, c);
    imprimir_campo(c);
  until ganhou(c) or perdeu(c);
  if perdeu(c) then
    writeln('Voce perdeu, tente novamente...')
  else
    writeln('Parabens, voce ganhou !!!');
  end.
```

Decisão 1: o que é um quadrado?

- Cada quadrado precisa guardar duas informações:
 - `info`: o conteúdo: `MINA=-1`, `BORDA=-2`, ou número de minas vizinhas (≥ 0)
 - `revelado`: se já foi aberto pelo jogador

```
quadrado = record
  info:      longint;    { BORDA, MINA ou n° de minas vizinhas }
  revelado:  boolean;    { false = escondido, true = visível }
end;
```

Decisão 2: como saber se o jogador ganhou

- Solução: guardar `falta_abrir` no TAD
 - Inicializado com `x*y - total_minas`
 - Decrementado a cada quadrado revelado
 - Quando chega a zero = vitória
- Custo: $O(1)$ por jogada

Decisão 3: borda sentinela

- Quando formos contar as minas vizinhas de um quadrado, vamos olhar os 8 vizinhos ao redor
 - Qual é o problema com os quadrados das bordas da matriz?

	0	1	2	3	4	5
0	B	B	B	B	B	B
1	B					B
2	B		X			B
3	B					B
4	B					B
5	B	B	B	B	B	B

Decisão 3: borda sentinela

- Quando formos contar as minas vizinhas de um quadrado, vamos olhar os 8 vizinhos ao redor
 - Qual é o problema com os quadrados das bordas da matriz?
- Matriz declarada como `array[0..MAX+1, 0..MAX+1]`
- Posições 0 e MAX+1 são borda: valor BORDA = -2
- Nenhum vizinho de borda vale MINA = -1
- Loop de vizinhos roda sem nenhum `if` de limite

Bloco 1 - Inicialização do campo

- Estrutura completa e constantes
- Visão geral da inicialização
 - `procedure inicializar_campo(var c: campo_minado);`
 - `procedure zerar_campo(var c: campo_minado);`
 - `procedure gerar_minas(var c: campo_minado);`

Bloco 1 - Inicialização do campo

- Visão geral da inicialização (continuação)
 - **function** num_vizinhos_com_mina(x, y: longint; c: campo_minado): longint;
 - **procedure** contar_vizinhos_com_mina(**var** c: campo_minado);

	j - 1	j	j + 1
i - 1	0	0	0
i	0	X	M
i + 1	0	0	0

Bloco 1 - Inicialização do campo

- Visão geral da inicialização (continuação)
 - **function** num_vizinhos_com_mina(x, y: longint; c: campo_minado): longint;
 - **procedure** contar_vizinhos_com_mina(**var** c: campo_minado);

	0	1	2
0	B	B	B
1	B	X	
2	B		

Bloco 2 - Impressão e leitura de jogada

- Estrutura completa e constantes
- Visão geral da inicialização
 - `procedure imprimir_campo(c: campo_minado);`
 - `procedure ler_jogada(var x, y: longint; c: campo_minado);`

Bloco 3 - Execução da Jogada (usa TAD Pilha)

- executar_jogada: três casos possíveis
 - Clicou em mina: derrota, revela tudo
 - Clicou em quadrado com vizinhos com mina (`info > 0`): revela só ele
 - Clicou em quadrado sem vizinhos com mina (`info = 0`): revela ele e toda a região conectada de zeros
 - Expansão em cadeia
 - Necessária estrutura para lembrar quais quadrados ainda precisam ser processados

Bloco 3 - Execução da Jogada (usa TAD Pilha)

- Adaptando o TAD Pilha para coordenadas
- A pilha do capítulo 12 era de longint. Aqui precisamos empilhar pares (x, y):

```
type
  coordenada = record
    x, y: longint;
  end;

  pilha = record
    ini: longint;
    v: array[1..TAM_MAX_PILHA] of coordenada;
  end;
```

Bloco 3 - Execução da Jogada (usa TAD Pilha)

- `function abrir_vizinhos_sem_mina(var c: campo_minado; var p: pilha): longint;`
- Detalhe crítico: quando marcar **revelado**

Bloco 4 - Funções de Status

- **function** ganhou (c: campo_minado): boolean;
- **function** perdeu (c: campo_minado): boolean;

Tudo que vocês aprenderam em um programa só

Conceito	Onde aparece
Registro	quadrado, campo_minado, coordenada
Matriz com borda	<code>array[0..MAX+1, 0..MAX+1]</code>
Sentinela	BORDA=-2 elimina verificações de limite
TAD	campo_minado com interface definida
Pilha de registros	abrir_vizinhos_sem_mina
Contagem incremental	falta_abrir evita $O(n^2)$ por jogada
Modularidade	programa principal com 10 linhas
Refinamentos sucessivos	estrutura inteira do capítulo

Departamento de Informática – UFPR

Algoritmos e Estruturas de Dados I

Prof.^ª Dr.^ª Mariane Cassenote

mariane@inf.ufpr.br

Refinamentos Sucessivos