

Projeto de algoritmos: Tentativa e Erro

ACH2002 - Introdução à Ciência da Computação II

Delano M. Beder

Escola de Artes, Ciências e Humanidades (EACH)
Universidade de São Paulo
dbeder@usp.br

09/2008

- Novidade: Nova técnica de solução de problemas.
- Vocês já aprenderam duas técnicas: (i) projeto por indução (incremental) e (ii) divisão e conquista.
- Tentativa e erro é uma técnica que utiliza *recursividade* (aliás, como também a técnica divisão e conquista utiliza).
- A recursividade pode ser usada para resolver problemas cuja solução é do tipo tentar todas as alternativas possíveis.
- Idéia para algoritmos tentativa e erro é decompor o processo em um número finito de subtarefas parciais.

Algoritmos Tentativa e erro

- Podem ser vistos como o processo de pesquisa ou de tentativa que gradualmente constrói e percorre uma árvore de sub-tarefas.

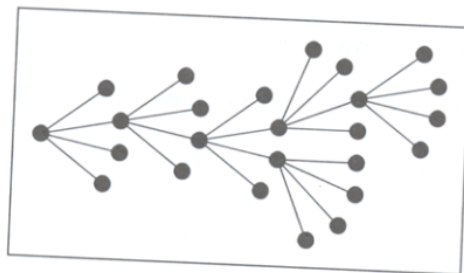
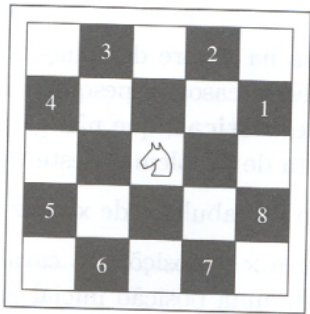


Figura 2.2 Árvore de subtarefas

Algoritmos Tentativa e erro

- Funcionamento dos algoritmos:
 - Passos em direção à solução final são tentados e registrados
 - Caso esses passos tomados não levem à solução final, eles podem ser retirados do registro.
- A busca na árvore de soluções pode crescer rapidamente (exponencialmente).

- Passeio do cavalo no tabuleiro de xadrez.
 - Dado um tabuleiro com $n \times n$ posições, o cavalo se movimenta segundo as regras do xadrez.
 - A partir de uma posição inicial (x_0, y_0) , o problema consiste em encontrar, se existir um passeio do cavalo com $n^2 - 1$ movimentos, visitando todos os pontos do tabuleiro uma única vez



```
procedimento tenta
BEGIN
  | inicializa seleção de movimentos |
  WHILE | movimento não bem sucedido | AND
    | existem candidatos a movimento | DO
    | seleciona próximo candidato ao movimento |
    IF | aceitável | THEN
      | registra movimento |
      IF | tabuleiro não está cheio | THEN
        | tenta novo movimento (chamada recursiva) |
        IF | não sucedido | THEN
          | apaga registro anterior |
        FI
      FI
    FI
  OD
END
```

- O tabuleiro \Rightarrow matriz $n \times n$.
- Situação de cada posição:
 - $t[x,y] = 0$, $\langle x,y \rangle$ não foi visitada
 - $t[x,y] = i$, $\langle x,y \rangle$ visitada no i -ésimo movimento, $1 \leq i \leq n^2$.
- As regras do xadrez são utilizadas para os movimentos do cavalo

```
public class KnightsTour {

    final int[] dx = { 2, 1, -1, -2, -2, -1, 1, 2 };
    final int[] dy = { 1, 2, 2, 1, -1, -2, -2, -1 };
    final int num;
    final int numSqr;
    int[][] table;

    public KnightsTour(int num) {
        this.num = num;
        this.numSqr = num * num;
        this.table = new int[num][num];
    }

    boolean isAcceptable(int x, int y) {
        boolean result = (x >= 0 && x <= num - 1);
        result = result && (y >= 0 && y <= num - 1);
        result = result && (table[x][y] == 0);
        return result;
    }

    ...
}
```

```
boolean tryMove(int i, int x, int y) {
    // Verifica a quantidade de movimentos
    boolean done = (i > numSqr);
    int k = 0;
    int u, v;

    while (!done && k < 8) {
        u = x + dx[k];
        v = y + dy[k];

        if (isAcceptable(u, v)) {
            table[u][v] = i;
            done = tryMove(i + 1, u, v); // tenta outro movimento
            if (!done) {
                table[u][v] = 0; // não sucedido. Descarta movimento
            }
        }
        k = k + 1;
    }
    return done;
}
```

```
void showTour(int x, int y) {
    table[x][y] = 1;
    boolean done = tryMove(2, x, y);
    if (done) {
        for (int i = 0; i < num; i++) {
            for (int j = 0; j < num; j++) {
                System.out.print(table[i][j] + " ");
            }
            System.out.println();
        }
    } else {
        System.out.println("No possible tour");
    }
}

public static void main(String[] args) {
    int n = Integer.parseInt(args[0]);
    int x = Integer.parseInt(args[1]);
    int y = Integer.parseInt(args[2]);
    new KnightsTour(n).showTour(x, y);
}
```

Resultado do exemplo

Resultado do exemplo

| | | | | | | | |
|----|----|----|----|----|----|----|----|
| 1 | 60 | 39 | 34 | 31 | 18 | 9 | 64 |
| 38 | 35 | 32 | 61 | 10 | 63 | 30 | 17 |
| 59 | 2 | 37 | 40 | 33 | 28 | 19 | 8 |
| 36 | 49 | 42 | 27 | 62 | 11 | 16 | 29 |
| 43 | 58 | 3 | 50 | 41 | 24 | 7 | 20 |
| 48 | 51 | 46 | 55 | 26 | 21 | 12 | 15 |
| 57 | 44 | 53 | 4 | 23 | 14 | 25 | 6 |
| 52 | 47 | 56 | 45 | 54 | 5 | 22 | 13 |

Resumo

- Foi apresentada a técnica de projeto de algoritmos:
 - tentativa e erro – tentar todas as alternativas possíveis.
- Foi apresentado um exemplo de algoritmos de tentativa e erro – passeio do cavalo no tabuleiro de xadrez.

Referência utilizadas: [1] páginas 42 a 46, [2] páginas 120 a 125.

[1] Nívio Ziviani. *Projeto de Algoritmos com implementações em C e Pascal*. Editora Thomson, 2a. Edição, 2004.

[2] Niklaus Wirth. *Algoritmos e Estrutura de Dados*. Editora LTC, Rio de Janeiro, 1989.