

A-gu-a

Tubarões e peixes numa batalha pela sobrevivência no planeta toroidal **A-gu-a**.

Algures numa direcção que só pode ser designada por recreação, e a uma distância só limitada pela capacidade de programação de cada um, o planeta **A-gu-a** vagueia entre as estrelas. A sua forma é a de um toro e é inteiramente coberto por água, os seus únicos cidadãos são tubarões e peixes, assim designados pela sua semelhança com as correspondentes criaturas terrestres. Os tubarões de **A-gu-a** comem os peixes, dos quais parece haver sempre uma população abundante.

Este texto que é uma adaptação do texto “Sharks and fish wage an ecological war on the toroidal planet Wa-tor” [1], descreve o ciclo de vida de duas populações (predadores e presas) em interacção num espaço limitado. O estudo desse tipo de situações é importante e daí vem a necessidade de construir programas que simulem essa situação.

Problema

Construa um programa que faça a simulação do ciclo da vida no planeta **A-gu-a**, sujeito ao seguinte conjunto de regras.

Planeta A-gu-a: O planeta **A-gu-a** tem a forma de um toro, a sua dimensão (representação plana, ver notas) máxima é de 20 linhas por 70 colunas.



Tempo: O tempo passa em intervalos discretos.

Regras referentes aos peixes: As regras referentes aos peixes dizem respeito à sua movimentação e à sua procriação.

Movimentação: A cada intervalo de tempo um peixe move-se aleatoriamente para uma das posições adjacentes livres. Se todas as posições possíveis já estão ocupadas o peixe não se move.

Procriação: Ao fim de um determinado período de tempo um peixe terá uma cria. O novo peixe ocupará a posição ocupada pelo peixe antes de este se mover, se não for possível o peixe mover-se ele também não procriará.

Regras referentes aos tubarões: As regras referentes aos tubarões dizem respeito à sua alimentação, movimentação, procriação, e à sua morte.

Alimentação/Movimentação: A alimentação tem precedência sobre a movimentação.

- A cada intervalo de tempo um tubarão seleccionará aleatoriamente uma posição adjacente que contenha um peixe, para a qual se deslocará, devorando o peixe.
- Não havendo peixes em nenhuma posição adjacente, o tubarão movimentar-se-á para uma delas de forma aleatória. Se todas as posições possíveis estiverem ocupadas por tubarões o tubarão não se move.

Procriação: Ao fim de um determinado período de tempo um tubarão terá uma cria. O novo tubarão ocupará a posição ocupada pelo tubarão antes de este se mover, se não for possível o tubarão mover-se ele também não procriará.

Morte: Se ao fim de um determinado período de tempo um tubarão não comer nenhum peixe, então o tubarão morre de fome.

O programa a desenvolver deve ter:

Como parâmetros de entrada: o número de linhas e colunas (`nlinhas`, `ncolunas`) da tabela que define a representação plana de **A-gu-a**; o número de peixes e de tubarões (`npeixes`, `ntubaroes`); o período de tempo para a procriação de peixes e tubarões (`pprocria`, `tprocria`); e o período de tempo para um tubarão morrer de fome (`tmorre`).

Como resultado (por iteração): A visualização da representação plana do planeta **A-gu-a**; o número de peixes e tubarões vivos nesse instante de tempo.

Como algoritmo

```
alg
  ler(dados)
  inicializa(A-gu-a)
  enquanto houver tubarões e peixes faz
    actualiza(A-gu-a)
    visualização(A-gu-a,npeixes,ntubaroes)
  fimenquanto
fimalg
```

Notas:

Condição de paragem: Se o sistema predador/presa em **A-gu-a** estabilizar, isto é, nenhuma das populações desaparecer então o ciclo principal do programa é um ciclo infinito, como tal, e embora não fazendo parte do problema, deve acrescentar um contador que assegure que o programa termina ao fim de um número determinado de iterações.

Gerador de números aleatórios: O procedimento de inicialização de **A-gu-a** recorre a um gerador de números aleatórios para colocar em posições aleatórias os peixes e os tubarões. Para cada peixe deve ser ainda inicializado aleatoriamente o espaço de tempo antes de procriar, valor entre 1 e `pprocria`. Para cada tubarão deve ser ainda inicializado aleatoriamente um espaço de tempo antes de procriar, assim como um espaço de tempo antes de morrer de fome, valores entre 1 e `tprocria` e `tmorre`, respectivamente.

Os procedimentos referentes à movimentação de peixes e tubarões recorrem ao mesmo gerador.

A seguinte função usa o gerador de números pseudo-aleatórios `drand48` para obter um número pseudo-aleatório $n \in \mathbb{N}$, com $a \leq n \leq b$.

```
function aleatorios(a,b: integer) : integer;

function drand48 : double ; asmname 'drand48'; external;
{ Função que devolve um real pseudo-aleatório no [0,1) }

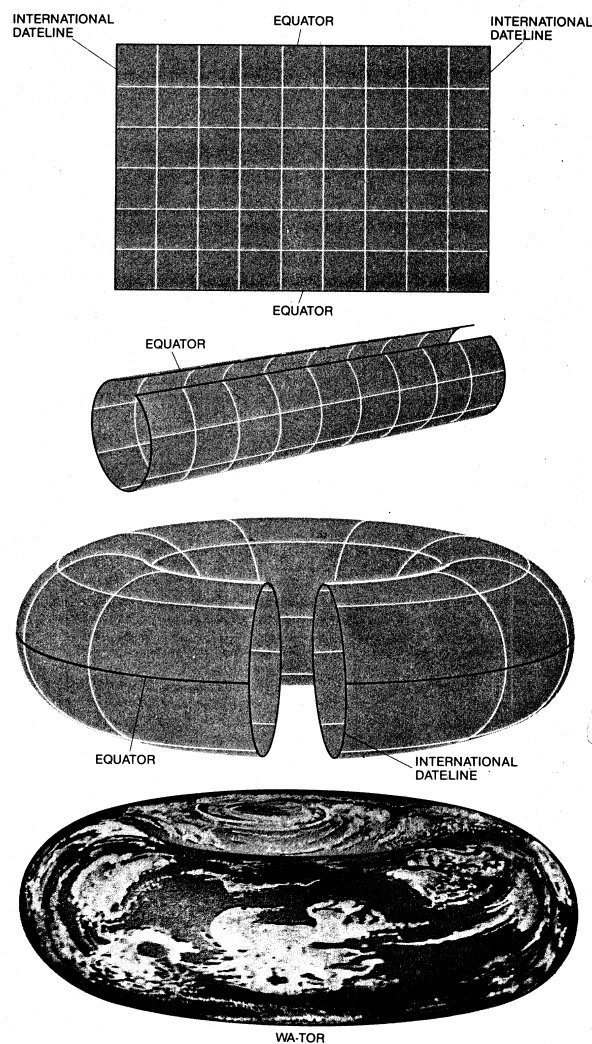
begin
  aleatorios:=round(a+drand48*(b-a))
end; { aleatorios }
```

Para que a sequência de números pseudo-aleatórios não se repita de uma execução do programa para outra usa-se a seguinte função:

```
function srand48(i : integer) : integer; asmname 'srand48'; external;
```

O valor inicial a ser dado a `srand48` é um outro parâmetro de entrada.

Representação plana de A-gu-a:



Relatório: Escreva um relatório, de acordo com o que foi descrito nas aulas teóricas. Descreva cuidadosamente a estrutura de dados que utilizou para implementar o planeta **A-gu-a**, assim como os peixes e os tubarões. O relatório deve incluir um exemplo de execução, o qual conterá a evolução numérica das populações de peixes e tubarões, esquecendo a visualização do planeta **A-gu-a**.

Nome do ficheiro: Na sua zona de trabalho, crie um directório **Projecto** e guarde o seu programa nesse directório sob o nome **proj.p**.

Identificação: Identifique o relatório e a listagem com o nome e “login” do(s) autor(es).

Data de entrega: 10 de Novembro de 2003.

Prazo de realização: até 12 de Dezembro de 2003.

Testes de aferição: de 15 a 18 de Dezembro de 2003.

Classificação: 0 a 3 valores.

Referências

- [1] A. K. Dewdney, Sharks and fish wage an ecological war on the toroidal planet Wa-tor, Computer Recreations.