



Universidade do Minho
Métodos de Programação 1
MIETI
Ano letivo 2015/2016

Projeto de grupo experimental – 3ª etapa

Problema Algorítmico – O Labirinto

Feito por: David Alves A79625
a79625@alunos.uminho.pt

Eduardo Araújo A72908
a72908@alunos.uminho.pt



Índice

Introdução.....	3
Enunciado do problema.....	4
Especificação algorítmica (executável no PortugolViana).....	5
Especificação do programa (executável em linguagem ANSI C).....	7
Análise da solução final proposta.....	9
Discussão crítica da solução final.....	10
Conclusões.....	11
Bibliografia.....	12



Introdução

Na nossa opinião, esta terceira e última fase do projeto experimental de Métodos de Programação 1 serviu para aplicarmos os conhecimentos que adquirimos durante o semestre à unidade curricular de Métodos de Programação 1 de uma forma mais prática/experimental e que engloba praticamente todos os conteúdos aprendidos a esta unidade curricular.

Consideramos também que ao realizar várias etapas de projetos experimentais pudemos aprender de uma forma mais clara e prática os conteúdos que aprendemos ao longo das várias aulas teóricas desta unidade curricular.

Sendo assim, esta terceira etapa do projeto era normal que tivesse um nível de dificuldade mais elevado, no entanto conseguimos concluir esta etapa e assim concluir o projeto experimental.



Enunciado do problema

Problema Algorítmico – o Labirinto

Um grupo de ratos de laboratório foi treinado para fugir de um labirinto. O labirinto é composto por células, podendo cada célula estar, ou não, ligada a outra célula. Entre cada duas células interligadas podem existir obstáculos, implicando um atraso temporal no percurso a executar entre as células interligadas. Algumas interligações só permitem percursos unidirecionais. Apenas uma das células do labirinto permite que os ratos saiam do mesmo.

Deve admitir-se que cada um dos ratos, depois de treinado num determinado labirinto, tem a capacidade de percorrer o caminho mais curto (ótimo) que lhe permite sair do labirinto, quando colocado aleatoriamente numa determinada célula.

Pretende-se realizar uma experiência que consiste no seguinte: é colocado um rato treinado em cada uma das células do labirinto; um cronómetro inicia uma contagem decrescente; quando o cronómetro chega a zero, conta-se o número de ratos que atingiu a saída do labirinto.

O problema em resumo: deve ler-se a descrição do labirinto e o valor inicial do cronómetro (entradas ou *inputs*) e calcular o número de ratos que atinge a saída do labirinto (saída ou *output*). Deve ainda assumir-se que cada célula pode albergar um número arbitrário de ratos.

Especificação das entradas: as células do labirinto estão numeradas sequencialmente de 1, 2, ..., N, em que **N** é o número total de células. Pode assumir-se $N \leq 100$. As primeiras três linhas de entrada contêm: **N**, o número de células do labirinto; **E**, o número da célula de saída do labirinto; **T**, o valor inicial do cronómetro

A quarta linha de entrada contém o número **M** de interligações do labirinto. Cada uma das seguintes **M** linhas especifica uma interligação à custa de três números inteiros: Dois números de células **A** e **B** (pertencentes ao intervalo 1, ..., N); O número inteiro de unidades temporais que demora a percorrer a interligação entre a célula **A** e **B**

Exemplo de entradas:

```
4
2
1
8
1 2 1
1 3 1
2 1 1
2 4 1
3 1 1
3 4 1
4 2 1
4 3 1
```

Especificação da saída (ou *output*): consiste numa única linha com um número inteiro que representa o número de ratos que atingiu a saída **E** do labirinto ao fim de **T** unidades de tempo. Exemplo de saída (tendo em conta as entradas indicadas no exemplo anterior):

```
3
```



Especificação algorítmica

inicio

```
inteiro vet [ 100 ] [ 3 ]
inteiro a, b, temp, m, tempo, cel, saida, ns2,z
inteiro k <- 1
inteiro linha <- 0
inteiro coluna <- 0
inteiro ns1 <- 0
inteiro ns <- 0
escrever "|-----|\n"
escrever "|","t","Entrada de dados"," ","\t"," |\n"
escrever "|-----|\n"
escrever "Numero de celulas:\n"
ler cel
escrever "Celula de saida:\n"
ler saida
escrever "Tempo inicial:\n"
ler tempo
escrever "Numero de interligacoes:\n"
ler m
//entrada dos dados

para linha de 1 ate m passo 1
  para coluna de 1 ate 3 passo 1
    escrever "ligacao A B TEMPO "
    ler vet [ linha-1 ] [ coluna-1 ]
  proximo
  escrever "-----\n" //separa as leituras dos ligações e respetivos tempos
proximo
para z de 1 ate m passo 1
  se vet [z-1] [1] = saida entao
    se vet [z-1] [2] <= tempo entao
      ns1 <- ns1 + 1
    fimSe
  fimSe
  se vet [z-1] [1] = vet [z] [0] E vet [z] [2] + vet [z-1] [2] <= tempo entao
    ns <- ns + 1
  fimSe
  escrever "\n"
proximo
//saida dos dados escritos da matriz
escrever "|-----|\n"
```



```
escrever "|","\\t","Ligacoes A B TEMPO"," ","\\t"," |\\n"
escrever "|-----|\\n"
para linha de 1 ate m passo 1
  para coluna de 1 ate 3 passo 1
    escrever " ", vet [ linha - 1 ] [ coluna - 1 ]
  proximo
  escrever "\\n"
proximo
ns2 <- ns + ns1 + 1
escrever "\\n"
escrever "Sairam ", ns2, " ratos"
fim
```



Especificação do programa (executável em linguagem ANSI C)

```
#include <stdio.h>

int main () {
    int a[50], b[50], t[50] ;
    int i, j, m, cel, saida, tempo, ns1=0, ns=0, ns2=0, ns3=0, z, vf;
    printf(" |-----|\n");
    printf(" |      Entrada de Dados      |\n");
    printf(" |-----|\n");
    printf("Numero de celulas:\n");
        scanf("%d",&cel);
    printf("Celula de saida:\n");
        scanf("%d",&saida);
    printf("Tempo inicial:\n");
        scanf("%d",&tempo);
    printf("Numero de interligacoes:\n");
        scanf("%d",&m);

    for ( i = 0; i < m; i++ ) {
        printf("Celula A:\n");
        scanf("%d",&a[i]);
        printf("Celula B:\n");
        scanf("%d",&b[i]);
        printf("Tempo:\n");
        scanf("%d",&t[i]);
        printf("\n");
    }
    printf("-----\n");
    printf("\n");
    printf(" |-----|\n");
    printf(" |      Ligacoes A B TEMPO      |\n");
    printf(" |-----|\n");
    for(z=0;z<m;z++){
        printf("%d",a[z]);
        printf("%d",b[z]);
        printf("%d\n",t[z]);
        printf("\n");
        if (b[z-1] != a[z]){
            if ( b[z] == saida && t[z] == 1 ){
                ns1++;}
            if ( b[z] == saida && 1 < t[z] <= tempo){
```



```
        ns3++;}  
}  
if (a[z]==b[z-1]){  
    if(t[z-1]+t[z]<=tempo){  
        ns++;}  
}  
ns2=ns1+ns+1+ns3;  
printf("Sairam %d ratos",ns2);  
  
return 0;  
}
```




Análise da solução final proposta

Em comparação com o algoritmo elaborado na 2ª etapa do projeto experimental, este algoritmo resolve mais situações problemáticas, no entanto não podemos afirmar que seja um algoritmo exemplar que resolve todas as situações problemáticas relacionadas com o problema proposto “o labirinto”, descrito na página 3 deste mesmo documento.

Sendo assim este algoritmo final elaborado para esta 3ª etapa do projeto resolve os casos em que os ratinhos chegam à célula de saída quando o tempo que demoram é igual a 1, mas também casos em que os ratinhos chegam à célula de saída por um conjunto de dois caminhos.

Como se pode observar nas seguintes imagens, para este programa, por acharmos que desta forma o programa está melhor organizado decidimos colocar no início uma caixa de texto com o título de Entrada de Dados enquanto o utilizador deste programa introduz os dados(input), e o título de ” Ligacoes A B TEMPO “, aquando da saída dos dados do programa(output). Depois de fazer a leitura de dados o programa guarda as ligações A B TEMPO num vetor bidimensional que chamamos de `vet[linha-1][coluna-1]`, através de um ciclo “para” linha que vai de 1 até m e passo 1(incrementa 1 a linha até chegar a m) onde através de a condição “se” verifica, se a célula B `vet[z-1][1]` é igual à saída e dentro desse “se” verifica se o tempo dessa situação em que a célula B é igual à saída, é menor ou igual ao tempo inicial do cronometro, e se assim for incrementa 1 a ns1. Depois o programa ainda verifica se a célula B da ligação que acaba de ser lida é igual à célula A da ligação seguinte(da linha seguinte) e ainda se a soma do tempo dessas duas ligações é menor ou igual ao tempo inicial do cronometro, e se esta condição se verificar incrementa 1 a ns.

No final o programa termina com uma matriz de m x 3 (sendo m o número de linhas) que mostra os dados das ligações A B TEMPO introduzidos e com uma frase, ”Sairam ns2 ratos”(sendo ns2 a soma de ns com ns1)

Exemplo:

```
|-----|
|  Entrada de dados  |
|-----|
Numero de celulas:
3
Celula de saida:
3
Tempo inicial:
4
Numero de interligacoes:
3
ligacao A B TEMPO 3
ligacao A B TEMPO 2
ligacao A B TEMPO 2
-----
ligacao A B TEMPO 2
ligacao A B TEMPO 3
ligacao A B TEMPO 2
-----
```

```
ligacao A B TEMPO 2
ligacao A B TEMPO 3
ligacao A B TEMPO 2
-----
ligacao A B TEMPO 1
ligacao A B TEMPO 3
ligacao A B TEMPO 5
-----
|-----|
|  Ligacoes A B TEMPO  |
|-----|
3 2 2
2 3 2
1 3 5
Sairam 3 ratos
```



Discussão crítica da solução final

Como já foi referido neste documento apesar de, comparativamente ao algoritmo que fizemos para 2ª etapa do projeto este algoritmo seja mais completo, no entanto ainda tem algumas limitações, como por exemplo, o facto de apenas resolver os casos em que existe uma célula B igual a uma célula A da linha seguinte ficando assim por resolver (apesar de inúmeras tentativas) os casos em que existe essa igualdade num maior intervalo de linhas.

Ao analisarmos e testarmos o programa concebido para este projeto foram apenas essas as falhas com que nos deparamos.

Em suma após todo este trabalho desenvolvido tanto na 2ª etapa como na 3ª etapa deste projeto experimental “O Labirinto” consideramos que fizemos um bom trabalho.



Conclusões

Concluimos assim este projeto experimental “O Labirinto”, que na nossa opinião foi um pouco difícil, no entanto conseguimos obter uma boa resolução algorítmica para este problema de programação.

Esperamos que esta experiência nos tenha servido para contribuir a termos uma melhor competência na resolução de problemas relacionados, não só com a programação em C mas também com outro tipo de linguagens de programação.



Bibliografia

- Anotações de Algoritmia fornecidas na *Blackboard* da unidade curricular;
- <http://www.cprogressivo.net/> ;
- Linguagem C - Luís Damas , LIDEL/FCA, 5a edição, 2001 ISBN: 978-972-722-156-1.