Estruturas de Dados e Algoritmos

Folha de exercícios número 10

2000/2001

Neste exercício[1] vai determinar a distância mínima entre dois pontos de um grafo dirigido. A cada ramo do grafo está associada uma distância não negativa. Estas distâncias estão numa matriz quadrada d de dimensões $|V| \times |V|$, sendo $d[a][b] =$`INFINITO` se $(a, b) \notin E$.

É conveniente, para testar o seu programa, considerar um grafo particular, inicializando para isso a correspondente matriz m.

1. **Método baseado na pesquisa em profundidade**
   Observações sumárias: Considere um programa que determine todos os caminhos simples entre 2 vértices (semelhante ao programa que foi apresentado nas aulas teóricas para a determinação de todos os ciclos simples e que se junta em anexo); nestes programas há retorno da função quando o vértice visitado está na pilha de vértices correntemente "activos" (em termos das chamadas à função). Se o vértice que se está a visitar for b (o vértice de chegada) compara-se o comprimento corrente c (desde desde o vértice de partida a), actualizando, se necessário, a distância mímina actual min. Por outro lado, se o comprimento corrente ultrapassar a distância mímima actual, efectua-se o retorno imediato da função (Porquê?). A descrição acima corresponde à utilização do método "branch and bound" na determinação de um valor óptimo.

2. **Tanto tempo!**
   Mostre através de um exemplo que o tempo de execução de um algoritmo de distância mínima entre 2 vértices que considere todos os caminhos simples entre esses vértices pode não ser polinomial em $|V|$.

3. **Por onde fui?**
   Complete o programa da pergunta 1 com vista a determinar também o (ou um) caminho correspondente à distância mínima.

4. **Método de Floyd**
   Com a mesma inicialização da matriz d determine a distância mínima entre os vértices a e b utilizando o algoritmo de Floyd.

5. **Por onde?**
   Complete o programa da pergunta 4 com vista a determinar também o (ou um) caminho correspondente à distância mínima.

6. **És rápido, Floyd?**
   Qual a eficiência do método de Floyd (em termos de$n = |V|$)?

[1] Sempre que necessário pode consultar os apontamentos "on line"; note todavia que o exame final é sem consulta.

```
/* Todos os ciclos ...  */
#define MAX 100
int n=6,
  ma[6][6]={{0,1... ...}}, sp, stack[MAX], comp[MAX]={0}, ncomp=0;
void df(int);/*---------------------------------------------*/
main(){
  int v,i;
  for(v=0;v<n;v++)
    if(!comp[v]){
      ncomp++;
      sp=0;
      df(v);
    }
}

int instack(int);
void printciclo(void);

/*---Pesq.profundidade--------------------------*/
void df(int v){
  int i;
  if(instack(v)){
    stack[sp++]=v;
    printciclo();
    sp--;
    return;
  }
  comp[v]=ncomp;
  stack[sp++]=v;
  for(i=0;i<n;i++){
    if(ma[v][i])
      df(i);
  }
  sp--;
}

/*--x em stack[0..sp-1]-------------------------*/
int instack(int x){
  int i;
  for(i=0;i<sp;i++)
    if(stack[i]==x)
      return(1);
  return(0);
}

/*--Imprime ciclo-------------------------------*/
void printciclo(void){
  int from=sp-2,i;
  while(stack[from]!=stack[sp-1])
    from--;
  for(i=from;i<sp;i++)
    printf("%3d",stack[i]);
  printf("\n");
}
```