

Semillero de Programación

Representación de grafos, getline y stringstream

Ana Echavarría Juan Francisco Cardona

Universidad EAFIT

15 de febrero de 2013

Contenido

- 1 Representación de grafos
- 2 Entrada usando getline y stringstream

Tarea

Tarea

Escribir un programa que genere las representaciones como matriz y lista de adyacencia de un grafo **no dirigido** G .

La primera línea de la entrada consiste de dos enteros n el número de nodos y m el número de aristas ($1 \leq n \leq 10$ y $1 \leq m \leq 100$). Las siguientes m líneas tienen cada una dos enteros u y v ($1 \leq u, v \leq n$) que indican que hay una arista que une los nodos u y v .

Solución para la matriz

```
1  using namespace std;
2  #include <iostream>
3
4  const int MAXN = 15;
5  int M [MAXN] [MAXN];
6
7  int main(){
8      int n, m;
9      cin >> n >> m;
10
11     for (int i = 0; i < m; ++i){
12         int u, v; cin >> u >> v;
13         u--; v--;
14         M[u] [v] = M[v] [u] = 1;
15     }
16     return 0;
17 }
```

Solución para la lista de adyacencia

```
1  using namespace std;
2  #include <iostream>
3  #include <vector>
4
5  const int MAXN = 15;
6  vector <int> g [MAXN];
7
8  int main(){
9      int n, m;
10     cin >> n >> m;
11
12     for (int i = 0; i < m; ++i){
13         int u, v; cin >> u >> v;
14         u--; v--;
15         g[u].push_back(v);
16         if (u != v) g[v].push_back(u);
17     }
18     return 0;
19 }
```

Limpieza de los contenedores

Limpiar los contenedores

Si en un problema se tienen múltiples casos de prueba, entre caso y caso hay que limpiar los contenedores porque estos están definidos como variable global.

Limpieza de la matriz

```
1  using namespace std;
2  #include <iostream>
3
4  const int MAXN = 15;
5  int M[MAXN][MAXN];
6
7  int main(){
8      int n, m;
9      while (cin >> n >> m){
10         for (int i = 0; i <= n; ++i){
11             for (int j = 0; j <= n; ++j){
12                 M[i][j] = 0;
13             }
14         }
15         // Crear el grafo
16     }
17     return 0;
18 }
```

Limpieza del vector

```
1  using namespace std;
2  #include <iostream>
3  #include <vector>
4
5  const int MAXN = 15;
6  vector <int> g [MAXN];
7
8  int main(){
9      int n, m;
10     while (cin >> n >> m){
11         for (int i = 0; i <= n; ++i){
12             g[i].clear(); //Limpiar la lista del nodo i
13         }
14         // Crear el grafo
15     }
16     return 0;
17 }
```

Matriz para grafos con pesos

En la representación como matriz, en lugar de almacenar un 1 en la posición u, v si los nodos u y v están conectados, almacenar el peso con el que están conectados el nodo u y el nodo v . Si los dos nodos no están conectados almacena $+\infty$.

$$M[u][v] = \begin{cases} W_{u,v} & \text{si } (u, v) \in E \text{ o} \\ +\infty & \text{en caso contrario.} \end{cases} \quad (1)$$

Lista de adyacencia para grafos con pesos

- En la representación como lista de adyacencia, si el nodo u está conectado con el v con un peso w , en $g[u]$ se almacena la pareja (v, w) .
- Para esto es útil el tipo de dato `pair <int, int>` que es una pareja de enteros.
- El grafo podría ser
`vector < pair <int, int> > g[MAXN].`
- Para conocer más del tipo de dato `pair` mirar <http://www.cplusplus.com/reference/utility/pair/?kw=pair>
- También se puede crear una estructura usando `struct`.

La función getline

Si la entrada es un texto y cada caso de prueba es una línea con del texto es necesario usar la función `getline` para leer esa línea de la entrada.

Un ejemplo de un problema donde se necesite usar `getline` para leer la entrada es el problema 483 - Word Scramble del juez <http://uva.onlinejudge.org>

Usando getline

```
1 using namespace std;
2 #include <iostream>
3
4 int main(){
5     string line;
6     while (getline(cin, line)){
7         // Do something with line
8     }
9     return 0;
10 }
```

El programa se ejecuta mientras que se pueda leer una línea del texto de entrada. La línea se lee y se almacena en el string line.

La clase stringstream

La clase `stringstream` sirve para manipular strings como si fueran una entrada.

Esta clase se utiliza cuando se tiene un string de entrada y se quiere extraer la información que tiene ese string separado por espacios.

Para utilizar la clase `stringstream` es necesario incluir `sstream`.

Usando stringstream

Ejemplo: Dado un texto de varias líneas diga cuántas palabras hay en cada línea.

```
1 using namespace std;
2 #include <iostream>
3 #include <sstream> // Necesario para usar stringstream
4
5 int main(){
6     string line;
7     while (getline(cin, line)){
8         stringstream ss(line); // El stream ss se crea con line
9         string word;
10        int count = 0;
11        while (ss >> word) count++; // ss se usa igual que cin
12        cout << count << endl;
13    }
14    return 0;
15 }
```

Tarea

Tarea

- 1 Registrarse en el sitio uva.onlinejudge.org
- 2 Entrar al sitio <http://contests.factorcomun.org/contests/49> y agregar su usuario de UVa con el botón *Add new team*
- 3 Resolver los 4 problemas de la competencia
- 4 El alumno que quede de primero en la competencia ganará un regalo sorpresa

Tarea

Pistas

Algunas pistas para los problemas

Problema A Usar el valor absoluto `abs(x)`.

Problema B Implementar el algoritmo descrito en el problema.

Problema C Utilizar el tipo de dato `double` para representar los números decimales.

Problema D Usar `getline` y luego `stringstream`.