1　アルゴリズムとデータ構造演習 第 4 回

1.1　注意

極端に遅いプログラムを提出した場合、正しく動作するものでもテストに失敗することがあります。そのようなプログラムはアルゴリズム的に不適切なので、改善して提出してください。

1.2　レポート課題

■課題 109　次のような形式で始点と終点のある重み付き無向グラフを定義することにする。

- 最初の行には、始点の名前が書かれている。
- 次の行には、終点の名前が書かれている。
- 以後は、各行が辺を意味している。辺は、両端点の名前と重みがタブを区切りとして並んだものである。
- 辺の重みは正整数で、同じ両端を持つ辺が複数存在することもある。

例えば、次のようなものがグラフの表現である。

```
A
F
A	B	3
A	C	1
A	E	12
B	A	5
B	C	1
B	D	8
C	D	10
C	E	9
D	F	2
E	F	3
```

辺の重みを 2 点間の移動にかかるコストと考えることにする。この表現でグラフを入力として受け取り、始点から到達可能な点で始点からの移動コストが最も低い点の名前を出力するプログラムを作成せよ。始点から到達可能な点が存在しない場合は、何も表示せず、該当する点が複数存在する場合は、それらのうちいずれかを表示すればよい。各点名の文字数は 99 以下、点の総数と辺の総数はそれぞれ 10000 以下と仮定してよい。

任意の点名を扱うことが困難な場合には、点名を非負整数に限定してもよい。ただし、その場合、レポート提出時のテストは最高で「2/3」である。

ヒント: 隣接リストを使ってグラフを扱う場合、例えば、以下のようなデータ構造を使うことができる。

```
/* 点のデータ型 */
typedef struct _vertex {
   char *name; /* 名前 */
```

```
   struct _edge *adj; /* 隣接リスト */
} vertex;

/* 辺のデータ型 */
typedef struct _edge {
   struct _vertex *sourcep; /* 始点へのポインタ */
   struct _vertex *targetp; /* 終点へのポインタ */
   int weight; /* 重み */
   struct _edge *next;
} edge;
```

グラフは vertex 型のデータの集合と考えられるので、 vertex の配列をグラフとして扱ったり、リストや二分木のためのフィールドを追加することで、 vertex へのポインタをグラフとして扱えばよい。

締切: 2011 年 01 月 10 日正午

■課題 110　課題 109 と同じグラフ表現を使うとする。グラフを入力として受け取り、始点から終点までの移動コストを最小にする経路を求め、その経路上の点を始点から順に終点まで表示するプログラムを作成せよ。始点から終点に到達可能でない場合は、「(no route)」と表示せよ。該当する経路が複数存在する場合には、それらのうち 1 つだけを表示するものとする。各点名の文字数は 99 以下、点の総数と辺の総数はそれぞれ 10000 以下と仮定してよい。

任意の点名を扱うことが困難な場合には、点名を非負整数に限定してもよい。ただし、その場合、レポート提出時のテストは最高で「2/3」である。

参考までに、少し大きなグラフのサンプルを配布しておく。このサンプルを入力として与えた場合の出力は以下の通りである。

渋谷
表参道
青山一丁目
永田町
麹町
市ケ谷
飯田橋
後楽園
本郷三丁目

締切: 2011 年 01 月 10 日正午

1.3　解説

一般に、グラフを表現する方法は単なる集合の場合に比べて多様です。グラフ表現の有名なものに、隣接行列や隣接リストを利用する方法があります。2 次元配列を使えば、直感的に隣接行列を実装することができます。また、これまでの課題で学んだ連結リストを使えば、隣接リストを実装することもできます。表現には、

それぞれにメリットとデメリットがあります。自分の実装したいアルゴリズムに応じて、データ構造を使い分けましょう。当然ですが、ヒントに書いてあるデータ構造だけが、実装の方法ではありません。他のよくある実装方法としては、頂点を自然数と同一視するというものがあります。この方法では、頂点の型は int で済みますが、その分、普通の整数と頂点を意味する整数を区別してプログラミングする必要があります。

課題 109 のヒントに掲載されているデータ構造は、 Dijkstra のアルゴリズムを使うことを想定しているわけではありません。 Dijkstra のアルゴリズムでは、各頂点に様々な情報を持たせる必要があるので、 vertex にフィールドを追加して情報を持たせるとよいでしょう。