

## 情報理工学演習Ⅲ 演習課題（第7回）

テーマ : グラフ

実施日 : 2020年6月4日（木）

担当 : 有村 博紀, 堀山 貴史, 金森 憲太朗 (TF), 加井 丈志 (TA)

授業連絡先 : enshuiii-csit@ist.hokudai.ac.jp (演習課題の提出・質問など)

担当連絡先 : kanamori@ist.hokudai.ac.jp (金森), kai@ist.hokudai.ac.jp (加井)

以下の各問に関して、それぞれプログラムを実装せよ。実装したコードは必ず、以下に示す提出様式を守って、**本授業のメールアドレス (enshuiii-csit@ist.hokudai.ac.jp) 宛にメールで提出すること。**

- メールの件名 : 「[csit ex3]xxxxxxxx」, ここで xxxxxxxx は自分の学生番号。
- 添付ファイル名 : 第 n 回演習の間 m に対するコードのファイル名は, 「n\_m.c」とする。

例 : 学生番号02180000の学生の場合	
• メール宛先	: enshuiii-csit@ist.hokudai.ac.jp
• メール件名	: [csit ex3]02180000
• 添付ファイル	: 7_1.c, 7_2.c, 7_3.c, 7_4.c

メール本文には、学生番号、所属コース、名前、実施回（第7回演習）、解いた問題番号を記載すること。課題提出によって出席をとるため、解いている途中の状態でも構わないので、演習時間内に一度は提出すること。

実装にはC言語を用い、gcc (ver.4.8.5 以降) でコンパイルすること。

コンパイル・実行例	
% gcc 7_1.c	// コンパイル
% ./a.out	// 実行

演習時間内に解き終わらなかった場合は、次回の演習時間までに提出すること。期間中であれば、何度でも再提出してよい。提出状況等は本授業のホームページまたはELMSグループ (moodle) 上に掲示予定なので、適宜確認すること。

問題番号に \* が付されたものは、発展課題である。解答は必須ではないが、解ければ成績評価に加点されるので、是非挑戦してほしい。

**【問1】** 以下の Sample 1 は、標準入力から下記の 入力形式 1 で頂点数  $n$ 、辺数  $m$  の無向グラフを受け取り、その隣接行列表現を出力するプログラムである。これを 7\_1.c として実装せよ。

Sample 1	
1: #include <stdio.h>	
2: #include <stdlib.h>	
3:	
4: int main() {	
5:     int n, m;	
6:     scanf("%d %d", &n, &m);	
7:     int** matrix = (int**) malloc(sizeof(int*) * n);	
8:     int i, j;	
9:     for (i = 0; i < n; i++) {	
10:         matrix[i] = (int*) malloc(sizeof(int) * n);	

```

11:  }
12:  for (i = 0; i < n; i++) {
13:      for (j = 0; j < n; j++) {
14:          matrix[i][j] = 0;
15:      }
16:  }
17:  int s, t;
18:  for (i = 0; i < m; i++) {
19:      scanf("%d %d", &s, &t);
20:      matrix[s][t] = 1;
21:      matrix[t][s] = 1;
22:  }
23:
24:  for (i = 0; i < n; i++) {
25:      for (j = 0; j < n; j++) {
26:          printf("%d ", matrix[i][j]);
27:      }
28:      printf("\n");
29:  }
30:  return 0;
31: }

```

```

n m          // n は頂点数, m は辺数
s0 t0
s1 t1
...
sm-1 tm-1    // 各 siti 間に辺が張られている (0 ≤ si, ti < n, 0 ≤ i < m)

```

入力形式 1

ただし、入力形式 1 は、以下を満たすとする：

- $s_0 \leq s_1 \leq \dots \leq s_{m-1}$ .
- 辺  $i, j$  ( $0 \leq i < j < m$ ) に対して、 $s_i = s_j$  ならば、 $t_i < t_j$ .

実装したプログラム 7\_1.c に以下の Input 1 を入力して実行し、結果を確認せよ。また、Input 1 が表す無向グラフを描画し、7\_1.c で得られた隣接行列表現と比較せよ（ただし、描画したグラフや比較結果は提出不要）。

```

7 8
0 4
0 6
1 2
1 3
1 4
1 6
3 4
3 5

```

Input 1

**【問2】** 頂点数  $n$ 、辺数  $m$  の無向グラフを受け取り、その隣接リスト表現を出力するプログラムを 7\_2.c として実装せよ。ここで隣接リストは、各辺情報が入力された順に、各頂点に対応するリストの先頭に要素を追加して構築すること。

入力は上記の 入力形式 1 の形式で標準入力から与えられるものとする。出力は各リストを降順に出力すること。各リストは各要素の直後に半角スペースを置いてこれらを区切り、各リストの出力の末尾には改行を入れること。

ヒント：隣接リストは連結リストの配列で表されることを思い出そう。連結リストの実装には、第2回演習で実装したソースコードが再利用できる。

入力例 1
7 8 0 4 0 6 1 2 1 3 1 4 1 6 3 4 3 5

出力例 1
6 4 6 4 3 2 1 5 1 3 1 0 3 1 0

入力例 1 は Input 1 と同じ無向グラフを表している。入力例 1 の隣接リスト表現（出力例 1）を、問1で描画した無向グラフや隣接行列表現と比較せよ（ただし、描画したグラフや比較結果は提出不要）。

実装したプログラム 7\_2.c に以下の Input 2 を入力して実行し、結果を確認せよ。

Input 2
8 10 0 1 0 6 1 3 2 3 2 4 3 5 3 6 4 6 4 7 5 7

**【問3】** 頂点数  $n$ 、辺数  $m$  の無向グラフ  $G$ 、及び  $G$  中の2つの頂点  $v_s$  と  $v_t$  を受け取り、 $G$  において  $v_s$  から  $v_t$  へ到達可能な最短路の長さを出力するプログラムを 7\_3.c として実装せよ。ここで、各頂点間の距離はすべて 1

とし、 $G$  には  $v_s$  から  $v_t$  に到達可能な路が少なくとも1つ存在することが保証されているものとする。  
入力は以下の 入力形式 2 の形式で標準入力から与えられるものとする。出力の末尾には改行を入れること。

$n\ m$ // $n$ は頂点数, $m$ は辺数 $s_0\ t_1$ $s_1\ t_1$ ... $s_{m-1}\ t_{m-1}$ // 各 $s_i t_i$ 間に辺が張られている ( $0 \leq s_i, t_i < n, 0 \leq i \leq m-1$ ) $v_s\ v_t$	入力形式 2
---	--------

実装したプログラム 7\_3.c に以下の Input 3 を入力して実行し、結果を確認せよ。

8 10 0 1 0 6 1 3 2 3 2 4 3 5 3 6 4 6 4 7 5 7 0 7	Input 3
---	---------

**【問4\*】**  $n+1$  個のマスが一直線上に並んでおり、各マスには左端から順に  $0, 1, 2, \dots, n$  の番号が振られている。この上で、スタート地点を番号  $0$  のマス、ゴール地点を番号  $n$  のマスとして、以下の手順で双六を行う。

1. $1 \sim 6$ の目が出るサイコロを振り、出た目の数だけ右のマスに移動する。 2. サイコロで移動した先のマスの番号 $x$ について、以下の操作を行う。 <ul style="list-style-type: none"><li>◦ <math>x</math> が <math>3</math> の倍数：2 マス左に戻る</li><li>◦ <math>x</math> が <math>4</math> の倍数：3 マス左に戻る</li><li>◦ <math>x</math> が <math>5</math> の倍数：3 マス右に進む</li><li>◦ それ以外      : そこに留まる</li></ul> 3. 最初の 1. に戻る。
---

番号が  $n$  以上のマスに移動することになった場合は、番号  $n$  のマスに移動するものとし、その時点で双六はただちにゴール（終了）とする。また、上記 2. における条件は重複して発動するものとする。たとえば、 $x = 12$  ならば 5 マス左に戻り、 $x = 60$  ならば 2 マス左に戻る。

1 以上の整数  $n$  を受け取り、上記の双六においてゴールするまでに必要な最小手数を出力するプログラムを 7\_4.c として実装せよ。但し、ここでいう手数とはサイコロを振った回数を指す。  
入力は標準入力から与えられるものとする。出力の末尾には改行を入れること。

実装したプログラム 7\_4.c に  $n = 4153$  を入力して実行し、結果を確認せよ。