

## Homework 4. Hasse Diagram Plotter

Shin Hong  
hongshin@handong.edu

## 1. Introduction

A *Hasse diagram* visually represents a partial order relation as a vertically arranged undirected graph. In this homework, you are asked to write a C program to draw a Hasse diagram for given information on a partial order relation. This assignment specifies the structure and the meaning of input data, and the expected properties of a Hasse diagram figure.

This homework requires both collaborative work and individual work. You are asked to collaborate with your team members on understanding the problem and devising algorithms. Besides, at implementing the program and writing the report, each of you must work individually and submit all results by **11:59 PM, 22 December (Wed)**.

## 2. Hasse Diagram

**Definition.** A Hasse diagram visualizes a partial order relation  $(S, <)$  by plotting the elements of  $S$  on a two-dimensional plane and connecting and arranging the elements with respect to the given relation  $<$ . For two different elements  $s_1$  and  $s_2$ , if  $s_1 < s_2$ ,  $s_1$  must be plotted at a lower position than  $s_2$ . And there must be a line connecting  $s_1$  and  $s_2$  if  $s_1 < s_2$  and there is no other  $s_3$  such that  $s_1 < s_3$  and  $s_3 < s_2$ .

**Input text data.** An input text file consists of multiple lines each of which contains two identifiers of elements such that the former one precedes the latter one. An element identifier is a word that consists of one to sixteen alphanumeric letters. Assume that there are at most 32 elements in a partial ordering relation.

Given binary relation in an input file, your program must consider its closure as the partial order relation to be plotted. Note that, for simplicity, the user may not give all pairs of a partial order relation, but part of these. The following is an example of input file that specifies the prerequisite course relation of a computer science curriculum (a blue underline character denotes a whitespace):

---

```
C_Java
Java_Java
C_DataStructure
LogicDesign_CompArch
C_OperatingSystem
DataStructure_EngProject
CompArch_EngProject
C_MobileApp
Java_MobileApp
```

---

**Output figure.** Given input text file, your program must generate a single .bmp file of a Hasse diagram figure. All elements must be plotted with their identifiers in the figure and properly arranged and connected according to the following rules:

- (1) an element  $s_1$  must be placed at a lower position to another element  $s_2$  if  $s_1 < s_2$ , and
- (2) a line connects two elements  $s_1$  and  $s_2$  if  $s_1 \neq s_2$  and  $s_1 < s_2$ , and there is no other element  $s_3$  such that  $s_1 < s_3$  and  $s_3 < s_2$

There is no restriction on image size, style, and decoration of a figure, if the figure is legible and adheres to the conventions.

Your program must use an open-source graphic library EZDIB for constructing .bmp file. EZDIB is portable that it consists of only two source code files, `ezdib.c` and `ezdib.h`, and has no dependencies other than the standard C libraries. You can find the source code files and an example at <https://github.com/hongshin/DiscreteMath/tree/master/assignments/ezdib> and descriptions on API functions at <https://www.codeproject.com/Articles/363908/Simple-two-file-graphics-library-for-C-Cplusplus>.

## 3. Requirements

**Functionalities.** Your program must identify and reject an invalid input file by showing an error message and not producing any image file. An input file is invalid if it violates any syntax or assumption, or it does not describe a partial order relation.

Your program must **receive one command-line argument** which specifies the file path of an input text file. For given input text `<file>`, your program must produce the image file as `<file>.bmp`. The following is the example of the command-line interface use:

---

```
$ls
hasse      input1.dat
$./hasse input1.dat
$ls
hasse      input1.dat      input1.dat.bmp
```

---

Your program must be written a C program. For generating graphical output, you are restricted to use EZDIB, and cannot use other libraries. The submitted code must be compatible with GCC 5.4.0 or higher versions. It may comprise of one or multiple C source code files. You must give a script to build an executable from the source code.

**Report writing.** Your report must be written in the given template, and it must not exceed 3 pages. In the evaluation, the descriptions on the following points are expected to be found from your report:

- an overview of the program design
- a description on how to analyze input
- a description on how to arrange elements
- demonstration of your program

In addition, you are recommended to write the findings, the challenges, unsolved questions, lessons learned, or any other interesting discussions related to this homework as well. Note that the evaluation is primary based on your report, and your implementation will be tested to check whether it consistently works as described in the report.

## 4. Submission Instruction

Upload a Zip file to Hisnet, containing all results including all files of your implementation and a PDF file of your report. Note that your report must be submitted as a PDF to avoid formatting and compatibility issues. The submission deadline is strict; no late submission will be accepted.