

A \LaTeX Template of Research Methodology Final Report Submission

A Research Report
Submitted in partial fulfillment of the requirements for the Degree of
Bachelor of Science in Computer Science and Engineering

Submitted by

Name 1	20xxxxxxxxxxxx
Name 2	20xxxxxxxxxxxx
Name 3	20xxxxxxxxxxxx

Supervised by

Mr./Ms. Supervisor Name

Official Designation
Department of Computer Science and Engineering
Southeast University, Bangladesh



Department of Computer Science and Engineering
Southeast University, Bangladesh

Dhaka, Bangladesh

Mon, YYYY

Letter of Transmittal

Mon, YYYY

The Chairman,
Department of Computer Science and Engineering
Southeast University, Bangladesh
Tejgaon, Dhaka

Through: Supervisor, Mr./Ms. Supervisor Name

Subject:

Dear Sir,

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Sincerely Yours,

Supervisor:

Name 1

20xxxxxxxxxxx

Name 2

20xxxxxxxxxxx

Name 3

20xxxxxxxxxxx

Mr./Ms. Supervisor Name

Official Designation & Supervisor

Department of Computer Science
and Engineering

Southeast University, Bangladesh

CANDIDATE'S DECLARATION

We, hereby, declare that the thesis presented in this report is the outcome of the investigation performed by us under the supervision of Mr./Ms. Supervisor Name, Official Designation, Department of Computer Science and Engineering, Southeast University, Bangladesh. The work was done through CSE459: Research Methodology course, in accordance with the course curriculum of the Department for the Bachelor of Science in Computer Science and Engineering program.

It is also declared that neither this research nor any part thereof has been submitted anywhere else for the award of any degree, diploma or other qualifications.

Name 1

20xxxxxxxxxxx

Name 2

20xxxxxxxxxxx

Name 3

20xxxxxxxxxxx

CERTIFICATION

This research titled, “**A ~~La~~X Template of Research Methodology Final Report Submission**”, submitted by the group as mentioned below has been accepted as satisfactory in partial fulfillment of the requirements for the degree B.Sc. in Computer Science and Engineering in Mon, YYYY.

Group Members:

Name 1	20xxxxxxxxxxxx
Name 2	20xxxxxxxxxxxx
Name 3	20xxxxxxxxxxxx

Supervisor:

Mr./Ms. Supervisor Name
Official Designation & Supervisor
Department of Computer Science and Engineering
Southeast University, Bangladesh

Shahriar Manzoor
Associate Professor & Chairman
Department of Computer Science and Engineering
Southeast University, Bangladesh

ACKNOWLEDGEMENT

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Dhaka

Mon, YYYY

Name 1

Name 2

Name 3

ABSTRACT

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Contents

LETTER OF TRANSMITTAL	i
CANDIDATES' DECLARATION	ii
CERTIFICATION	iii
ACKNOWLEDGEMENT	iv
ABSTRACT	v
List of Figures	viii
List of Tables	ix
1 Introduction to this template	1
1.1 How to use this template	1
2 Tutorial on setting the Research Details	2
2.1 Information Folder	2
2.2 Prologue Folder	3
3 Tutorial on Using Pictures	4
3.1 Single Figure	4
3.2 Multiple Figure	4
3.3 Using .svg for Vector Graphics	5
3.4 Multiple .svg Figures	5
4 Tutorial on Algorithm writing using <i>algorithm2e</i> package	6
4.1 Referring functions and algorithms	6
4.2 Writing a function	6
4.3 Writing an algorithm	6
5 Using Codes & Scripts	9
5.1 C++ Code	9
5.2 Python Code- having only a part of the code	10

5.3	Java Code- referring a code	10
5.4	Matlab Code	12
6	Tables	14
6.1	Text paragraph in table	14
6.2	Long table	15
7	Citation Tutorial	18
7.1	Including references	18
7.2	Bibilography style	18
7.3	references.bib File	18
7.4	How to add references in .bib file	19
7.5	Citing a website	20
8	Clever reference	21
	References	23
A	Extra Algorithm for Appendix	24
B	Extra Codes for Appendix	25

List of Figures

3.1	Demo for single image	4
3.2	General caption.	5
3.3	Demo for .svg image	5
3.4	General caption.	5
7.1	Paper description in google scholar page.	19
7.2	Getting citation bibtex codes	19
7.3	Copying bibtex source	19
7.4	Populating bibtex source	20
8.1	Demo for single image	21

List of Tables

6.1	Sample Table	14
6.2	Table with no paragraph	14
6.3	Table with paragraph	15
6.4	Feasible triples for a highly variable Grid	15

Chapter 1

Introduction to this template

1.1 How to use this template

1. Open an account in www.overleaf.com
2. To practice and understand how the template works, go to new project->upload project and up “SEU_Research_Final_Template.zip” file
3. Go through main.tex to see how the files are structured.
4. In parallel, read the given “SEU_Research_Final_Template.zip.pdf” and try to tune and change the commands.
5. After you get used to the template, you can start editing the template.
6. Start working on your research and contact us for any problem you are facing regarding this template.

Chapter 2

Tutorial on setting the Research Details

The preamble of your research book will contain the cover page, abstract of the research and table of contents. In these parts, several things (the title of the research, name of the students and the supervisor etc.) will appear multiple times. So, we have set up an arrangement so that you can write these details or parameters once and have them written in different places at once.

2.1 Information Folder

Information folder in the template contains several .txt files consisting of the research details.

date.txt This file contains the month and year of the research book. Change it and compile the project to see the changes in cover page.

students.txt This file contains the name and IDs of the students. There are one name per line. Follow the name with the student ID. Name and ID must be separated by a comma. Change the names and IDs and compile the template to see the changes accordingly in cover page.

supervisor.txt This file contains the name and the designation of the Supervisor of the research group. Name and designation will be separated by a comma. Change them and compile the project to see the changes in cover page.

title.txt This file contains the name of the research. Change it and compile the project to see the changes in cover page.

worktype.txt This file contains the type of your work (research). Ignore it for now.

2.2 Prologue Folder

Prologue folder contains large texts to be changed- the likes of abstract. There is **abstract.tex** file where you can write the corresponding texts in \LaTeX and see the changes in abstract page.

Chapter 3

Tutorial on Using Pictures

3.1 Single Figure

This demo is for setting up a single .jpg/.png image. See the corresponding source code to see how Figure 3.1 is used and referred to.



Figure 3.1: Demo for single image

3.2 Multiple Figure

This demo is for setting up multiple .jpg/.png images side by side (subfigures). See the corresponding source code to see how Figure 3.2a and Figure 3.2b is used and referred to. Figure 3.2a and Figure 3.2b are parts of Figure 3.2.



(a) First



(b) Second

Figure 3.2: General caption.

3.3 Using .svg for Vector Graphics

This demo is for setting up a single .svg image. .svg is used for Vector graphics- stretching or shortening does not affect the details of the images. See the corresponding source code to see how Figure 3.3 is used and referred to. See, you do not need to include .svg extension as filename in code.

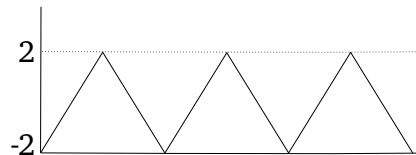
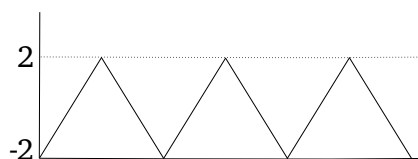


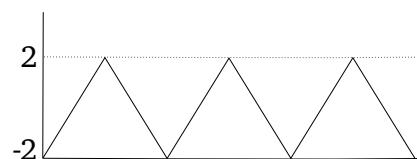
Figure 3.3: Demo for .svg image

3.4 Multiple .svg Figures

This demo is for setting up multiple .svg images. See the corresponding sourcecode to see how Figure 3.4a and Figure 3.4b are used and referred to. They are parts of Figure 3.4.



(a) First



(b) Second

Figure 3.4: General caption.

Chapter 4

Tutorial on Algorithm writing using *algorithm2e* package

4.1 Referring functions and algorithms

Use `setminus ref` command preceded with **Function** or **Algorithm** to refer them. Labeling them has been shown in the algorithm's source code. Function 1, algorithm 1- see the latex code to see how to refer the functions and algorithms.

4.2 Writing a function

\LaTeX files under Algorithms folder show how to write an algorithm/function in \LaTeX and refer an algorithm/a function/ a line/ a block. Use `algorithm2e` package for writing algorithms. You do not need to include it, it has already been done. You can write `algorithm.tex` in root folder too. But we kept the algorithms separately in another folder for modularizing our code.

Sourcecode for Function 1 is shown in Listing 4.1.

4.3 Writing an algorithm

Example of writing algorithm is shown in Algorithm 1.

```

1 %MARGIN FOR ALGORITHM
2 \IncMargin{1em}
3 %START OF FUNCTION

```



```

4 \begin{func}
5 \begin{algorithm}[!p]
6 %%%SETTING KEYWORDS FOR STYLING THEM IN THE FUNCTION%%
7 %%%
8 %\SetKwData{x}{y} sets a data/datastructure y to be referred as x in the
   algorithm. See where subbst has been used in sourcecode and
   SliceUpperBoundBST in the algorithm.
9 %%%
10
11 \SetKwData{subbst}{SliceUpperBoundBST}
12 %%%
13 %\SetKwFunction{x}{y} sets a function y to be referred as x in the algorithm.
   See where scanslab has been used in sourcecode and ScanSlab in the
   algorithm.
14 %%%
15
16 \SetKwFunction{scanslab}{ScanSlab}
17
18 %%%
19 %Usage of Input,Output, Data( Persistent data like database)
20 %%%
21 \KwInput{ A set of slices $S_{slice}$ }
22 \KwOutput{Whatever}
23 \KwData{$xyz$}
24 %%%LINEGAPE
25 \BlankLine
26 %%%FOR LOOP BLOCK
27 \For{ \upshape{each} $s_i$ \upshape{in} $S_{slice}$ }
28 {
29     %%% \gets means assignment
30     $s_i.R$ \gets \text{the set of rectangles currently intersecting with } s.
       i$ \;
31     $(s_i.S_{slabs}, g_{\maxub})$ \gets $ \scanslab{$s_i.R$} \;
32     $\subbst.update(s_i.id, g_{\maxub})$ \;
33     $s_i.p_{\{c\}}$ \gets null$ \;
34     $s_i.lazy$ \gets false$ \;
35     $s_i.maxregsearched$ \gets false $ \;
36 }
37 %%%CAPTIONING THE FUNCTION
38 \caption{PrepareSlices$(S_{slice})$}
39 \label{func:prepslice}
40 %END OF FUNCTION
41 \end{algorithm}
42 \end{func}
43 %MARGIN FOR ALGORITHM
44 \DecMargin{1em}

```

Listing 4.1: Sourcecode for Function 1

Function 1: PrepareSlices(S_{slice})**Input:** A set of slices S_{slice} **Output:** Whatever**Data:** xyz

```

1 for each  $s_i$  in  $S_{slice}$  do
2    $s_i.R \leftarrow$  the set of rectangles currently intersecting with  $s_i$ ;
3    $(s_i.S_{slabs}, g_{maxub}) \leftarrow \text{ScanSlab}(s_i.R)$ ;
4    $\text{SliceUpperBoundBST.update}(s_i.id, g_{maxub})$ ;
5    $s_i.p_c \leftarrow \text{null}$ ;
6    $s_i.lazy \leftarrow \text{false}$ ;
7    $s_i.maxregsearched \leftarrow \text{false}$ ;

```

Algorithm 1: SolveCMaxRS⁺ ($e^+(o_e), a, b, p_c^*$)**Input :** An $e^+(o_e)$ event, query size $a \times b$, and current maximal point p_c^* **Output:** Updated maximal point p_c^*

```

1  $r_e \leftarrow$  the  $a \times b$  rectangle centered at  $o_e$ ;
2  $QTree.insert(\text{new Node}(r_e))$ ;
3  $S_e \leftarrow$  set of slices intersecting  $r_e$ ;
4  $S_{lazy} \leftarrow$  set of slices marked lazy;
5  $isPrunable \leftarrow \text{true}$ ;
6 for each  $s_i \in S_e$  do
7   if after the addition  $R \cup r_e$  conforms to  $X$  then
8      $s_i.lazy \leftarrow \text{true}$ ;
9      $isPrunable \leftarrow \text{false}$ ;
10     $S_{lazy} \leftarrow S_{lazy} \cup \{s_i\}$ ;
11 if  $isPrunable = \text{true}$  then
12   return  $p_c^*$ 
13 PrepareSlices( $S_{lazy}$ );
14  $p_c \leftarrow \text{SliceSearchMR}(p_c^*)$ ;
15 return  $p_c^*$ 

```

Chapter 5

Using Codes & Scripts

Listing environment is used for including codes. Here we have included some C++, Matlab, Java, Python etc. codes as examples. Both full codes and snippets have been shown as example. Also, we showed how to refer a code.

5.1 C++ Code

```
1 #include <bits/stdc++.h>
2
3 using namespace std;
4
5 #define lol long long int
6 #define MOD 1000000007
7 #define MAX 1005
8
9 lol dp[MAX+7][MAX+7];
10
11 void build_dp ()
12 {
13     dp[0][0] = 1;
14
15     for(int i=0; i<MAX; i++)
16         for(int j=0; j<MAX; j++)
17             {
18                 dp[i][j+2] += dp[i][j];
19                 dp[i+1][j+1] += dp[i][j];
20                 dp[i+2][j+2] += dp[i][j];
21                 dp[i+1][j] += dp[i][j];
22                 dp[i][j+2] %= MOD;
23                 dp[i+1][j+1] %= MOD;
24 }
```

```

25         dp[ i+2][ j+2] %= MOD;
26         dp[ i+1][ j ] %= MOD;
27     }
28 }
29 int main()
30 {
31     build_dp();
32     srand(time(0));
33     int x,y,N = 20000;
34     FILE *in = fopen("1.in","w");
35     FILE *out = fopen("1.out","w");
36     while(N--)
37     {
38         x = rand()%1000+1;
39         y = rand()%1000+1;
40         fprintf(in,"%d %d\n",x,y);
41         fprintf(out,"%lld\n",dp[x][y]);
42     }
43     //fprintf(in,"%d %d",0,0);
44     fclose(in);
45     fclose(out);
46     return 0;
47 }

```

Listing 5.1: C++ example

5.2 Python Code- having only a part of the code

```

1 import helper
2 from helper import *
3
4 """
5 Scheduling functions
6 """
7
8 def do_linear_schedule(t):

```

Listing 5.2: Python code (partial) example

5.3 Java Code- referring a code

Here Listing 5.3 is a Java code. See the Latex Source to learn how to refer.

```

1 /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates

```

```
4  * and open the template in the editor.
5  */
6
7  package syncthread;
8
9  import java.util.logging.Level;
10 import java.util.logging.Logger;
11
12 /**
13  *
14  * @author student
15  */
16 public class PrinterThread implements Runnable{
17
18     public static int a = 0;
19     synchronized public void print(int n)
20     {
21         try {
22             for (int i=0;i<n;i++){
23                 a ++ ;
24                 System.out.println("A "+a+": printing "+i+ " in thread "+Thread
25 .currentThread().getName());
26                 Thread.sleep(100);
27             }
28         } catch (InterruptedException ex) {
29             Logger.getLogger(PrinterThread.class.getName()).log(Level.SEVERE,
30 null, ex);
31         }
32     }
33     @Override
34     public void run() {
35         print(500);
36         //throw new UnsupportedOperationException("Not supported yet."); //To
37 change body of generated methods, choose Tools | Templates.
38     }
39
40     public static void main(String args[])
41     {
42
43         PrinterThread pthread1 = new PrinterThread();
44
45         Thread t1 = new Thread(pthread1, "Pthread1");
46         Thread t2 = new Thread(pthread1, "Pthread2");
47
48         t1.start();
```

```
48     t2.start();
49
50
51 }
52 }
```

Listing 5.3: Java code for referencing sample

5.4 Matlab Code

```
1 IM=imread('sunderland.jpg'); % Read in a image
2 whos
3 figure;
4 subplot(4,2,1);
5 imshow(IM); % Display image
6 title('Input image');
7 FF = fft(IM); % Take FFT
8 IF = uint8(((FF)));
9 subplot(4,2,2);
10
11 imshow(IF);
12 title('Fourier of Input image:complex representation (real part)');
13
14 whos
15 subplot(4,2,3);
16 IF = uint8(abs(FF));
17
18 imshow(IF);
19 title('Fourier of Input image:modulus representation');
20
21 whos
22
23 subplot(4,2,4);
24
25 imshow(angle(FF));
26 title('Fourier of Input image:argument representation');
27
28 whos
29
30 IFF = ifft(FF); % take IFFT
31 whos
32 FINAL_IM = uint8(IFF); % Take real part and convert back to UINT8
33 whos
34 subplot(4,2,5);
35
```

```
36 imshow(FINAL_IM);  
37 title('Inverse Fourier of Input image fourier:whole');  
38  
39 IFF = ifft(abs(FF));  
40 FINAL_IM = uint8(IFF);      % Take real part and convert back to UINT8  
41 whos  
42 subplot(4,2,6);  
43  
44 imshow(FINAL_IM);  
45 title('Inverse Fourier of Input image fourier:modulus');
```

Listing 5.4: Sample Matlab Code

Chapter 6

Tables

This <https://www.tablesgenerator.com/> site is a great tool for generating latex tables.

Here, table 6.1 is a multirow, multicolumn merged cells table. You can visit that site, use your data and bold, center, give border- infact can do any kind of styling on your table. Remember, sometimes it may show in the generated code that you need to include

```
{\usepackage{multirow}}
```

in preamble. You don't need to do so- we have done it already. **Remember, in tables, we give captions above them, unlike the figures.**

Table 6.1: Sample Table

ID	Assignment		Result	
	Mid	Total	Mid	Total
1	3	4	5	6
2	2	3	4	6

6.1 Text paragraph in table

Use pwidth instead of l/c/r in tabel environment to get wrapped text. Table 6.2 shows a nonwrapped text containing table and Table 6.3 shows a wrapped text containing one.

Table 6.2: Table with no paragraph

A quick brown fox jumps over the lazy dog	A lion does not bother itself with a sheep's opinion
---	--

Table 6.3: Table with paragraph

A quick brown fox jumps over the lazy dog	A lion does not bother itself with a sheep's opinion
---	--

6.2 Long table

[We took help from <http://users.sdsc.edu/ssmallen/latex/longtable.html> for this example.]

The longtable package can help you out for showing a table spanning multiple pages. It allows you to

1. specify the column headings such that it prints on each page.
2. add a caption on each continued page to indicate that it the table is continued from the previous page.
3. you can add a footer to indicate that a table will be continued on the following page

The longtable syntax is identical to the regular table environment. The Table 6.4 spans more than one page:

Table 6.4: Feasible triples for highly variable Grid, MLMMH.

Time (s)	Triple chosen	Other feasible triples
0	(1, 11, 13725)	(1, 12, 10980), (1, 13, 8235), (2, 2, 0), (3, 1, 0)
2745	(1, 12, 10980)	(1, 13, 8235), (2, 2, 0), (2, 3, 0), (3, 1, 0)
5490	(1, 12, 13725)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
8235	(1, 12, 16470)	(1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1, 0)
10980	(1, 12, 16470)	(1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1, 0)
13725	(1, 12, 16470)	(1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1, 0)
16470	(1, 13, 16470)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
19215	(1, 12, 16470)	(1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1, 0)
21960	(1, 12, 16470)	(1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1, 0)
24705	(1, 12, 16470)	(1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1, 0)
27450	(1, 12, 16470)	(1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1, 0)
30195	(2, 2, 2745)	(2, 3, 0), (3, 1, 0)
32940	(1, 13, 16470)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
35685	(1, 13, 13725)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
Continued on next page		

Table 6.4 – continued from previous page

Time (s)	Triple chosen	Other feasible triples
38430	(1, 13, 10980)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
41175	(1, 12, 13725)	(1, 13, 10980), (2, 2, 2745), (2, 3, 0), (3, 1, 0)
43920	(1, 13, 10980)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
46665	(2, 2, 2745)	(2, 3, 0), (3, 1, 0)
49410	(2, 2, 2745)	(2, 3, 0), (3, 1, 0)
52155	(1, 12, 16470)	(1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1, 0)
54900	(1, 13, 13725)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
57645	(1, 13, 13725)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
60390	(1, 12, 13725)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
63135	(1, 13, 16470)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
65880	(1, 13, 16470)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
68625	(2, 2, 2745)	(2, 3, 0), (3, 1, 0)
71370	(1, 13, 13725)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
74115	(1, 12, 13725)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
76860	(1, 13, 13725)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
79605	(1, 13, 13725)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
82350	(1, 12, 13725)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
85095	(1, 12, 13725)	(1, 13, 10980), (2, 2, 2745), (2, 3, 0), (3, 1, 0)
87840	(1, 13, 16470)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
90585	(1, 13, 16470)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
93330	(1, 13, 13725)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
96075	(1, 13, 16470)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
98820	(1, 13, 16470)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
101565	(1, 13, 13725)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
104310	(1, 13, 16470)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
107055	(1, 13, 13725)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
109800	(1, 13, 13725)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
112545	(1, 12, 16470)	(1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1, 0)
115290	(1, 13, 16470)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
118035	(1, 13, 13725)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
120780	(1, 13, 16470)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
123525	(1, 13, 13725)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
126270	(1, 12, 16470)	(1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1, 0)
129015	(2, 2, 2745)	(2, 3, 0), (3, 1, 0)
131760	(2, 2, 2745)	(2, 3, 0), (3, 1, 0)
134505	(1, 13, 16470)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
Continued on next page		

Table 6.4 – continued from previous page

Time (s)	Triple chosen	Other feasible triples
137250	(1, 13, 13725)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
139995	(2, 2, 2745)	(2, 3, 0), (3, 1, 0)
142740	(2, 2, 2745)	(2, 3, 0), (3, 1, 0)
145485	(1, 12, 16470)	(1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1, 0)
148230	(2, 2, 2745)	(2, 3, 0), (3, 1, 0)
150975	(1, 13, 16470)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
153720	(1, 12, 13725)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
156465	(1, 13, 13725)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
159210	(1, 13, 13725)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
161955	(1, 13, 16470)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
164700	(1, 13, 13725)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)

Chapter 7

Citation Tutorial

This chapter shows how you can add references of publications and website urls and refer them.

7.1 Including references

In **main.tex**, you will find a line

```
\input{Extensions/bibliography.tex}
```

. **Don't delete it.** In the template given, the line will be there to include the references list.

7.2 Bibilography style

In **main.tex** , you can change the style of bibliography by uncommenting one of the following lines.

```
\bibliographystyle{ieeetr}  
% \bibliographystyle{acm}
```

They represent ieee transaction and ACM style respectively.

7.3 references.bib File

The project template will have a references.bib file. You will add your references there.

7.4 How to add references in .bib file

Steps:

1. Go to scholar.google.com and find the paper.
2. Click in the ending quotation sign beneath the paper description as shown in Figure 7.1.

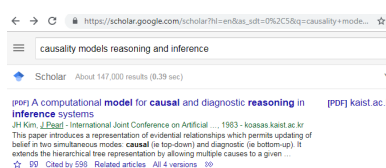


Figure 7.1: Paper description in google scholar page.

3. Click on “Bibtex” as shown in Figure 7.2.

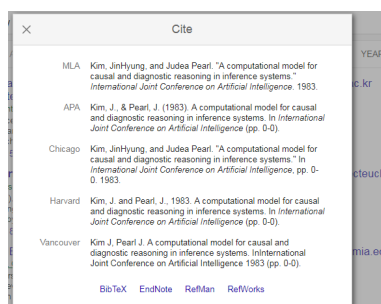


Figure 7.2: Getting citation bibtex codes

4. Copy the whole code. Notice the arrowed line in Figure 7.3. You will need this to refer publications.

```
@inproceedings{kim1983computational,
  title={A computational model for causal and diagnostic reasoning in inference systems},
  author={Kim, Jinhung and Pearl, Judea},
  booktitle={International Joint Conference on Artificial Intelligence},
  pages={0-0},
  year={1983}
}
```

Figure 7.3: Copying bibtex source

5. Paste the code in references.bib file as shown in Figure 7.4.
6. Use this kind of command for referring.

```
\cite{kim1983computational}
```

See, how [1] is referred here.

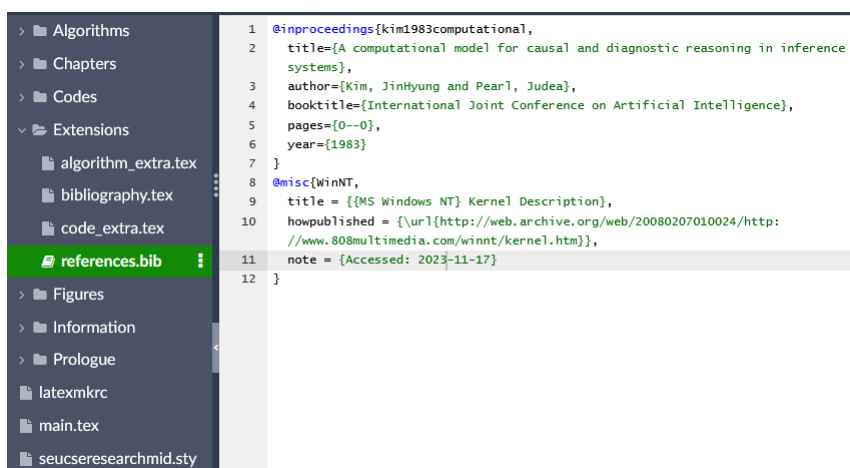


Figure 7.4: Populating bibtex source

7.5 Citing a website

Here is how an entry of website in references.bib file looks like.

```

@misc{WinNT,
title = {{MS Windows NT} Kernel Description},
howpublished = {\url{http://web.archive.org/web/20080207010024/
http://www.808multimedia.com/winnt/kernel.htm}},
note = {Accessed:2010-09-30}
}

```

Use

```
\cite{WinNT}
```

To refer this example site. See how we [2] referred it. You can use this sample bibtex code to refer any site- just change the label after **misc**, **title**, the site link in the bracket after **url** and last accessed time in note.

Chapter 8

Clever reference

You can use `\Cref` and `\cref` for clever referencing. If you do so, you do not need to specify if the referred item is a chapter or table or figure. See how we refer this logo of SEU in different styles.



Figure 8.1: Demo for single image

1. Using

Figure~\ref{fig:seulogo}

Figure [8.1](#)

2. Using

\Cref{fig:seulogo}

Figure [8.1](#)

3. Using

\cref{fig:seulogo}

fig. [8.1](#)

References

- [1] J. Kim and J. Pearl, “A computational model for causal and diagnostic reasoning in inference systems,” in *International Joint Conference on Artificial Intelligence*, pp. 0–0, 1983.
- [2] “MS Windows NT kernel description.” <http://web.archive.org/web/20080207010024/http://www.808multimedia.com/winnt/kernel.htm>. Accessed: 2023-11-17.

Appendix A

Extra Algorithm for Appendix

Function 2: PrepareSlices(S_{slice})

Input: A set of slices S_{slice}

Output: Whatever

Data: $x y z$

```

1 for each  $s_i$  in  $S_{slice}$  do
2    $s_i.R \leftarrow$  the set of rectangles currently intersecting with  $s_i$ ;
3    $(s_i.S_{slabs}, g_{maxub}) \leftarrow \text{ScanSlab}(s_i.R)$ ;
4    $\text{SliceUpperBoundBST.update}(s_i.id, g_{maxub})$ ;
5    $s_i.p_c \leftarrow \text{null}$ ;
6    $s_i.lazy \leftarrow \text{false}$ ;
7    $s_i.maxregsearched \leftarrow \text{false}$ ;

```

Appendix B

Extra Codes for Appendix

```
1 #include <bits/stdc++.h>
2
3 using namespace std;
4
5 #define lol long long int
6 #define MOD 1000000007
7 #define MAX 1005
8
9 lol dp[MAX+7][MAX+7];
10
11 void build_dp ()
12 {
13     dp[0][0] = 1;
14
15
16     for(int i=0; i<MAX; i++)
17         for(int j=0; j<MAX; j++)
18             {
19                 dp[i][j+2] += dp[i][j];
20                 dp[i+1][j+1] += dp[i][j];
21                 dp[i+2][j+2] += dp[i][j];
22                 dp[i+1][j] += dp[i][j];
23                 dp[i][j+2] %= MOD;
24                 dp[i+1][j+1] %= MOD;
25                 dp[i+2][j+2] %= MOD;
26                 dp[i+1][j] %= MOD;
27             }
28 }
29
30 int main ()
31 {
32     build_dp ();
33     srand (time (0));
```

```
33  int x,y,N = 20000;
34  FILE *in = fopen("1.in","w");
35  FILE *out = fopen("1.out","w");
36  while(N--)
37  {
38      x = rand()%1000+1;
39      y = rand()%1000+1;
40      fprintf(in,"%d %d\n",x,y);
41      fprintf(out,"%lld\n",dp[x][y]);
42  }
43  //fprintf(in,"%d %d",0,0);
44  fclose(in);
45  fclose(out);
46  return 0;
47 }
```

Listing B.1: C++ example

Generated using Research L^AT_EX Template, Version 1.0. Department of Computer Science and
Engineering, Southeast University, Bangladesh.

The Original Version was Developed by Tashreef Muhammad on Friday 17th November, 2023

This report was generated on Friday 17th November, 2023 at 17:17:07.