FINAL YEAR PROJECT AND RESEARCH PROJECT GUIDELINES

NOTE

This document is evolving. It will be subjected to revisions with a view to help the current and future students learn and enjoy more from their FYP experience. If you think of relevant points, welcome to let me know and I'll add in your observations.

Some part of the document is useful to research students also.

Revised: 14th November, 2018

Contributors: Dr. S.Y. Yuen and FYP students

Table of Contents

GOALS

1. To learn to handle a large technical project.
2. To learn good project management skills.

PREPARATION

1. Buy a log book and bring it every time you have meeting with me. Record your project progress, calculations, results of discussions etc. in the log book for your personal reference.
2. Familiarize with the assessment method used in EE
3. Contact other FYP students using the phone contact. Form discussion groups.
4. Read “How to run successful projects”, F.O’Connell, available from library. (Either 1st or 2nd edition ok. If you wish to, you may skip "Ch. 3 Step 3 – There must be one leader", "Ch. 4 Step 4 – Assign people to jobs", "Ch. 6 Step 6 – Use an appropriate leadership style" and "Ch. 14 Picking the right people" on a first reading.)

FIND USEFUL KNOWLEDGE ABOUT THE PROJECT

FYP students have to do a literature survey to find updated information about their research topic. Such information is contained in conferences and journal papers. Conferences are short publications of 4 to 8 pages. Journals are long publications describing more mature and in depth results. Useful bird eye views can be obtained through resources on the web such as Wikipedia.

Please read “Useful Database and Web Resources for Searching Information” in my web page.

DOING THE PROJECT

1. Identify the project objectives:
    a) There may be one or more project objectives.
    b) The objective(s) should be precise, clear, achievable, and well defined.
    c) The list of objectives should be presented in point form.
    d) Consult me for advice.
2. Write a clear technical specification.
3. Identify the equipment and critical resources needed. Decide on the programming language(s) you would use.
4. Identify milestones. Milestones should be clear, concrete, demonstrable achievements.
5. Prepare a project planning schedule. Hand it in to me.
6. Have periodic meetings with me. You should take the initiative to arrange meetings. Revise items 1-5 as the project progresses.

Note: Project objectives, technical specifications, equipment lists etc. can be cut and pasted into your final report.

AMOUNT OF EFFORT REQUIRED

The fyp is a 6 credit elective spread across two semesters. For coursework, each credit usually takes 2 hours /week (e.g. 18 credits /semester $\Rightarrow$ 36 hours in total per week. For comparison, a full time staff in a company works 44 hours per week including lunch time.) So it is expected that you spend an average of at least 6 hours /week on the project.

Since there is now an assessment towards the end of the first semester, it is advised to have a good time management and spread the project progress evenly in the two semesters, i.e., it is better to work constantly at the project, spending about 6 hours per week each week for the whole year.

LITERATURE SURVEY AND MILESTONES

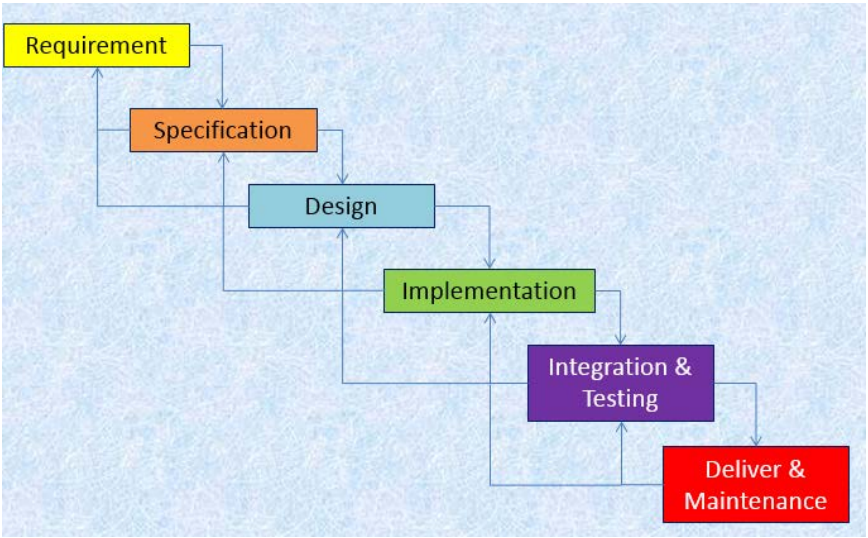
1. To understand what the project is about, you should do a literature survey of related ideas and methods.
2. This is the literature survey section in your fyp report. The survey should be written like the introduction of a journal paper. Consult my research paper to learn how a survey is done. A completed survey can simply be cut and pasted into your report.
3. Your survey should comment on existing works related to your problem. For each approach, do the following as best as you can:
   a) describe briefly how the approach works;
   b) describe the application of the approach, report numerical figures, e.g. accuracy, speed, etc. as provided in the paper;
   c) comment on the advantages of the approach;
   d) comment on the limitations of the approach.
4. Before you decide on the milestones, you need to read some journal and conference papers to find useful information. Then decide on the exact method to be used. When looking at those papers, you may find them difficult to read since you do not have the necessary background. Generally, read the abstract and conclusion, decide whether it is relevant before reading on. Consult me for advice.
5. When deciding on the method to be used. Consider the following criteria:
   a) Is it a recent method? The more recent the better.
   b) How accurate is it? Higher the accuracy the better.
   c) How fast is it? Faster the better.
   d) Is it beyond your ability to implement it successfully? Choose a method that you are confident you can finish.
   e) Are there novel ideas you generate such that you can further investigate and expand on? (This is relatively minor).

FYP MEETINGS

1. We have bi-weekly regular meetings.
2. You are welcome to have other impromptu meetings with me. The responsibility to arrange other meetings lies with you. You should take the initiative to arrange meetings with me.
3. When you attend/arrange a meeting, I expect that there is an agenda of items to discuss. Normally, I would expect that a milestone is achieved and you demo the results, hand in the survey, etc. A meeting with no agenda is a waste of time.
4. However, there may be other types of meeting. This happens when you need clarification of directions, encounter an obstacle that you cannot solve, cannot find critical equipment, etc. Don't be afraid to call a meeting of this nature, but inform me the problem before the meeting.

FYP PLANNING CHART

1. You need to submit a FYP planning chart.
2. The planning chart should identify the milestones and assign the estimated date for achieving them.
3. A milestone is a concrete event that one can use to demonstrate progress. A common mistake in the FYP planning chart is that the milestones are vague and one cannot tell whether things have been done or not after reaching the deadline (e.g. study a method). You should identify and write down concrete milestones (e.g. implement and debug adding sound recording to an app).
4. The SMART principle used in sports (Specific, Measurable, Achievable, Relevant, Time-bound) can be used to define milestones.
5. Milestones may also be defined following the waterfall model.



6. It is a good idea to break down the project into many milestones. By doing so, one can get a good idea of what needs to be done, and whether the project is feasible. A project is infeasible if some milestones are found to be vague or un-achievable. Also, by achieving milestones in turn, one gets the feeling that more and more are accomplished, and boasts confidence in completing the project. There are many ways to solve problems. Breaking down into milestones is a technique known as decomposition.
7. Some makeup time should be allowed in your planning. Project may and often delay.
8. I shall require students to conduct a demonstration around the end of the first semester. You are expected to demonstrate to me something concrete that has

been achieved. For example, if you have learnt objective C from week 1 to 12, that is not something concrete and will receive no mark. However, writing a simple program in objective C that implements part of the project is concrete progress.

PROGRAMMING

1. Form the good habit of commenting your programs as you write them.
2. Read the document “How to write good programs: Structured Programming” in my web page.
3. For each function, under the function name, include a comment section that has the following:

    a) description of the function
    b) description of the input variables
    c) description of the output variables
    d) example of the use of the function
    e) version: this includes the date at which the function is last modified
    f) known bugs

4. MATLAB has excellent help and extensive libraries and toolboxes. It is extremely handy to test out a thought or ideas. You can write short programs and test out very quickly. You can also do symbolic mathematics and numerical optimization using MATLAB toolboxes. It is also a good tool for statistical analysis of your results. The Signal Processing laboratory has licenses for MATLAB as well as an extensive range of toolboxes.
5. The disadvantages of MATLAB are 1) if you can't eliminate the FOR loops, your program will become intolerably slow to run; 2) you have to do it in the laboratory.
6. One possibility is to use the *resize* function to resize images to a small size for initial testing. Then re-implement in C.
7. C /C++ is much quicker if your program needs nested FOR loops.
8. If you are writing C/C++ program, C/C++ implementations of numerical optimization algorithms can be found in the books "Numerical recipes in C" and "Numerical recipes in C++". These two books (with CD ROM) are available in the library. Use them as they provide numerically stable implementation.

USING SOURCE CODE OF OTHERS

It is permitted to use other people's source code. However, in all occasions (e.g. report, demo etc.), you MUST acknowledge the use. Moreover, you should make sure that the source code is correct. If it is not, your whole system will not work, with wrong and unpredictable results. This would of course ruin your whole project.

One solution may be to use the source code as reference and write your own; another is to thoroughly understand and verify the source code before you make it your own, with the above acknowledgment.

FOLLOWING PROGRAMS WRITTEN BY OTHERS

Stage 1: Get to know the program.

1. Backup the original source code.
2. Try to compile and run once the code.
3. Find the main function of the program, trace the flow and briefly record the structure of the program.

Stage 2: Understand the code.

1. If there is any class or structure defined, look first.
2. Record down the meanings of parameters. E.g.: x= 0 : disable, =1 :enable,
3. In C, those structure or parameters always defined in header file (*.h files).
4. See the function one by one. It is better to understand the basic, subroutine functions first (i.e. use a bottom up approach).
5. Add comments after you understand the code.
6. Give a brief description of each function. E.g.: The input, process, output.

Stage 3: Modify the code

1. When you need to modify the whole function, it is better that you make a copy of the original function as backup.
2. Add remarks on the lines you changes. E.g.: When you modify, for what, what is the original one.
3. It is better to test the function once when you finish modifying every small part, as it is very hard to debug when many parts of the function has been modified.
4. In debugging, it is good idea to use “printf” to print out the state (E.g. parameters, flags …); you will get better understand what is going on in the program.

NUMERICAL and SYMOBLIC OPTIMIZATION

1. Computers can do
   i) numerical optimization : constrained optimization, root finding etc.
   ii) symbolic optimization : solving partial differential equations, partial difference equations, simplifying complicated expression, etc
2. MATLAB, Mathematica and Maple can do numerical and symbolic optimization.
3. If you are writing C/C++ program, C/C++ implementations of numerical optimization algorithms can be found in the books "Numerical recipes in C" and "Numerical recipes in C++". These two books (with CD ROM) are available in the library. Use them as they provide numerically stable implementation.

DEBUGGING

If you find that your program has bugs, you can:

1. Compare your current codes with the previous backed up one to find the differences made. This can minimize the area to find out the bug.
2. Use a set of synthetic data to calculate the intermediate results for every stage. Use breakpoints to check if there is any difference with the results that you think should be correct to locate the bug.

WEB SERVER

If need arises, you can rent a web server for your fyp projects and share amongst yourselves.

Please remember to pass the access information (e.g. password) and renewal details (e.g. when the web page needs to be renewed) to the next batch of fyp students before you finish the project.

BACKUP

Important: Always backup your work after each modification. Murphy's Law states that anything that can go wrong, will, and at the time when you most needed it not to go wrong. So beware. Two real experiences: The hard disk of an MPhil student failed when he was writing his thesis! Similarly, the hard disk of an undergraduate student crashed just before his final year project presentation, and he has no backup!

CHANGE OF FYP TITLE

You can change your project title.

Welcome to change it to a more appropriate title. The title should be informative, concise and reflects the achievement of your project. For advice on choosing titles, please refer to 英文科學論文寫作 by R.M. Lewis et al. (Chinese edition copyright @2007 by Jong Wen Books Co.).

WRITING FYP ABSTRACT

When writing fyp abstract, this is one possible structure:

1. State the objective and significance of the work.
2. Give a succinct summary of your methodology.
3. Give a summary of your experimental findings.
4. Give a conclusion based on your findings.
5. Give recommendations (optional)

Example 1

T. Jalbert , M. Jalbert , K. Hayashi , "State rankings of cost of living adjusted faculty compensation", *Accounting and Taxation,* vol. 1, no. 1, pp.121-137, 2009.

Abstract:
"In this paper we rank states based on higher education faculty compensation. Data on 574 universities across each of the 50 states and the District of Columbia are aggregated to develop a state compensation average. The analysis examines states both on a raw basis and on a cost of living adjusted basis. Rankings are reported for various academic classifications of faculty. Rankings based on salary data alone and salary and benefit combined data are presented. The results indicate that rankings of states based on raw and cost of living adjusted data are markedly different. The results suggest that faculty seeking employment opportunities should carefully consider cost of living issues. Administrators should design salary packages that reflect the local cost of living conditions in their area to attract quality faculty."

Structure:
1. "In this paper we rank states based on higher education faculty compensation."

2. "Data on 574 universities across each of the 50 states and the District of Columbia are aggregated to develop a state compensation average. The analysis examines states both on a raw basis and on a cost of living adjusted basis. Rankings are reported for various academic classifications of faculty. Rankings based on salary data alone and salary and benefit combined data are presented."

3. "The results indicate that rankings of states based on raw and cost of living adjusted data are markedly different."

4. "The results suggest that faculty seeking employment opportunities should carefully consider cost of living issues."

5. "Administrators should design salary packages that reflect the local cost of living conditions in their area to attract quality faculty."

Variations are of course possible. This is another example:

Example 2:

S.Y. Yuen, C.K. Chow, “A Genetic algorithm that adaptively mutates and never revisits,” *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 454-472, Apr. 2009.

Abstract:
"A novel genetic algorithm is reported that is non-revisiting: It remembers every position that it has searched before. An archive is used to store all the solutions that have been explored before. Different from other memory schemes in the literature, a novel binary space partitioning tree archive design is advocated. Not only is the design an efficient method to check for revisits, if any, it in itself constitutes a novel adaptive mutation operator that has no parameter. To demonstrate the power of the method, the algorithm is evaluated using 19 famous benchmark functions. The results are as follows: a) Though it only uses finite resolution grids, when compared with a canonical genetic algorithm, a generic real-coded genetic algorithm, a canonical genetic algorithm with simple diversity mechanism, and three particle swarm optimization algorithms, it shows a significant improvement. b) The new algorithm also shows superior performance compared to Covariance Matrix Adaptation Evolution Strategy (CMA-ES), a state of the art method for adaptive mutation. c) It can work with problems with large search spaces with dimensions as high as 40. d) The corresponding CPU overhead of the binary space partitioning tree design is insignificant for applications with expensive or time consuming fitness evaluations, and for such applications, the memory usage due to the archive is acceptable. e) Though the adaptive mutation is parameter-less, it shows and maintains a stable good performance. However, for other algorithms we compare, the performance is highly dependent on suitable parameter settings."

Structure:

1. "A novel genetic algorithm is reported that is non-revisiting: It remembers every position that it has searched before."

2. "An archive is used to store all the solutions that have been explored before. Different from other memory schemes in the literature, a novel binary space partitioning tree archive design is advocated. Not only is the design an efficient method to check for revisits, if any, it in itself constitutes a novel adaptive mutation operator that has no parameter."

3, 4. "To demonstrate the power of the method, the algorithm is evaluated using 19 famous benchmark functions. The results are as follows: a) Though it only uses finite resolution grids, when compared with a canonical genetic algorithm, a generic real-coded genetic algorithm, a canonical genetic algorithm with simple diversity mechanism, and three particle swarm optimization algorithms, it shows a significant improvement. b) The new algorithm also shows superior performance compared to Covariance Matrix Adaptation Evolution Strategy (CMA-ES), a state of the art method for adaptive mutation. c) It can work with problems with large search spaces

with dimensions as high as 40. d) The corresponding CPU overhead of the binary space partitioning tree design is insignificant for applications with expensive or time consuming fitness evaluations, and for such applications, the memory usage due to the archive is acceptable. e) Though the adaptive mutation is parameter-less, it shows and maintains a stable good performance. However, for other algorithms we compare, the performance is highly dependent on suitable parameter settings."

ENGLISH

1. Read "The Elements of Style" by William Strunk and E.B. White. This is a very good book if you wish to write good English.
2. 英文科學論文寫作 by R.M. Lewis et al.(Chinese edition copyright @2007 by Jong Wen Books Co.) is an excellent book that teaches you how to write a research paper. The material is also useful for writing fyp report.
3. Use MS Word spell and grammar check before handing in any document.
4. Use MS Word Thesaurus to find the synonyms of a word. This is useful for learning new usages and alternative descriptions.
5. Have with you a good English-Chinese dictionary. The Longman English-Chinese Dictionary of Contemporary English is recommended.
6. Form a habit of reading English books. Personally, I like reading science fictions. I recommend reading s.f. books by I. Asimov. He wrote beautifully and his stories are always interesting. You can read short or long stories.
7. The departmental guideline is that reports with bad English will NOT receive A grade.
8. Really good reports (A- or above) will be selected to participate in the Outstanding Academic Papers by Students (OAPS) award. This will be good item to include in your CV. So try your best effort in good writing.

TIPS ON PROJECT PRESENTATION

General

1. Different people have different presentation style and ways of organizing a talk. The following are suggestions only. You may do it in your own way as long as you present it well.

2. Have a good model of your audience. One very common mistake is to assume that your audience is as familiar as you about the background. Usually they are not. Assume your audience is intelligent but do not have the background. It is important to explain the background and motivation clearly.

3. Your presentation should be self-contained. No symbols should be left unexplained. You should introduce a concept the first time it appears.

4. You are probably the person who is most interested in the topic you are going to present. Don't expect the audience to be as interested as you; instead, it is your job to make your presentation enthralling.

Preparing your presentation

1. In your first slide, show the title of your project, your name, your supervisor and assessor's name.
2. In your second slide, show a clear objective or list of objectives. Make sure that they are clear, in point form and well defined.
3. Two slides on the motivation and background of the project.

   Motivation : *Why* the project is important / worth doing.

   Background : What past works by others have been done on the problem.

   Do not have too many slides on it, as you would not have enough time to talk about your work.
4. The bulk of the presentation is what you have done in the project. Divide it into

   a) Theory /Methodology:

   - *How* it is done. The assessment panel consists of Professors who may not be working in the field. You must present your work i) technically of a high standard; ii) clearly understood by non-specialists who are intelligent.

   b) Experimental Results :

   - *What* has been done.

- Include the full experimental setup, e.g. what machine, language, control parameters, threshold settings, …
- Data - Organize your data into tables, graphs, images.
- Make use of animation in power point to present and highlight your data and findings.
- If possible, include a demo, which would make a more attractive presentation. There are two ways to prepare a demo.
  a) *live demo* - This will gain many marks if the demo works, but Murphy's law states that if a thing can go wrong, it would. Therefore if you decide to do a live demo, make sure that it would work. Also prepare a recording as backup.
  b) *recording* - This is a safe alternative to doing a live demo. You record a live demo and play back. Then you have to make sure that Murphy's law does not strike down your video recorder :)

5. Use a slide to conclude. Your conclusions should be in point form. Each point should clearly states what you have achieved/found.
   It is a good idea to think about the question: "Why should we give the project an A?" and try to answer this question convincingly in your conclusion.
6. The last slide is a slide "Q and A".
7. It is a good idea to prepare transparencies which provides more detailed information. You may also think of possible questions and prepare transparencies which answer them. These transparencies need not be shown due to lack of time. However, if the question is asked, it gives an impression that you are well prepared.
8. Check the grammar of your power point. Make sure that the English is good and there is no spelling mistake.
9. Rehearse your presentation beforehand. It is a good idea to rehearse it with fellow fyp students. Ask them to have pencil and paper in hand while listening to your presentation and give you some comments afterwards. It is also a good idea to listen to good presentations by fellow students and check out how a presentation is run beforehand.
10. There is an assessment form used by Panel Members. It may be found in the fyp home page. Find out the assessment criteria and the rubic used.

The Presentation

1. Load your software into the computer before the beginning of the session.
2. Find a way such that after the last one finishes, you can get your power point running immediately.
3. Be calm and composed during your presentation. Do not speak too softly or too loudly.
4. Formal dressing is needed. Marks will be deducted if you appear unprofessional.
5. Keep to the time. Don't over-run. Rehearse beforehand with your friends and check the time. You may also video tape your rehearsal to see how well you are doing.
6. Speak fluent English. If your English is weak, rehearse more. Don't read from a paper.
7. Some students may like to bring a deck of small cards. If so, only jot down the main points; and maintain eye contact. Personally, I prefer to use the main points on the transparencies and elaborate rather than use cards.
8. Eye contact should be maintained with the audience.
9. Enjoy the presentation! You seldom have chance to speak before Professors and your fellow classmates. Treat it as a valuable learning experience. A cheerful person would give a better presentation.

Q and A

1. Listen carefully. Make sure you understand what the question is before answering. The assessors find it annoying that if you are not answering their question.
2. If really in doubt, put the question in your own words and ask them whether it is really the question they are asking.
3. When faced with a difficult question, you can buy a little time by saying "This is a good question", but you have to give them a good answer right after that.
4. If you really can't answer the question, say so. People find it annoying if you pretend to answer the question, but actually don't.
5. Be polite. Have good manners and etiquette.

DEMONSTRATION

Demonstration is very important.

Respect Murphy's Law (http://en.wikipedia.org/wiki/Murphy%27s_Law). Prepare well and well in advance. Unexpected things may happen, especially for hardware projects. So it is important to have a good preparation and a backup plan.

Before the demonstration, arrive much earlier to the laboratory. Do not let your supervisor /assessor wait while you set up. It would give a very bad impression.

Set everything up and make sure that everything is running properly. Check and re-check. If you have time, it is better to prepare a video as backup, just in case the demo cannot be done for some unexpected reasons. Then you can show the video and ask for a second chance to sort out the unexpected problem.

FYP PROGRESS REPORT

1. You need to submit a progress report.

2. Take the opportunity to prepare an excellent progress report as it may be possible to re-use some parts of the progress report (e.g. the survey) in your final report.

FORMAT FOR QUOTING REFERENCES

Follow the IEEE format. Examples are given below:

Conference Paper:

[1] T. Bäck, "The interaction of mutation rate, selection and self-adaptation within a genetic algorithm," in *Parallel Problem Solving from Nature*, R. Manner and B. Manderick, Eds. Amsterdam, The Netherlands: North-Holland, 1992, vol. 2, pp. 85-94.

[2] C.W. Sung, S.Y. Yuen, "On the analysis of the (1+1) evolutionary algorithm with short-term memory," in *Proc. IEEE Congress on Evolutionary Computation* (CEC), June 2008, pp. 235-241.

[3] T. Friedrich, N. Hebbinghaus and F. Neumann, "Rigorous analyses of simple diversity mechanisms," in *Proc. Genetic and Evolutionary Computation Conf.* (GECCO), July 2007, pp. 1219-1225.

Paper within an edited book:

[4] M. Kimura, "Overdevelopment of the synthetic theory and the proposal of the neutral theory," in *The Neutral Theory of Molecular Evolution*. Cambridge, MA: Cambridge University Press, 1983, ch. 2, pp. 25-33.

Journal Paper:

[5] G. Rudolph, "Convergence analysis of canonical genetic algorithms," *IEEE Trans. Neural Netw.*, vol. 5, no. 1, pp. 96-101, 1994.

[6] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. on Evolutionary Computation*, vol. 1, no. 1, pp. 67-82, 1997.

[7] A.E. Eiben, R. Hinterding, and Z. Michalewicz, "Parameter control in evolutionary algorithms", *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 124-141, 1999.

Book:

[8] J.H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: Univ. Michigan Press, 1975.

[9] F. Glover and M. Laguna, *Tabu search*, Kluwer Academic Publishers, 1997.

Follow the exact format as shown above. For additional examples, look at the references in any paper of IEEE Transactions on Evolutionary Computation.

Check that all references follow the same, consistent format. It will look unprofessional if otherwise.

When quoting references, make sure that all details are correct. Check it over two times. Incorrect references are unprofessional will give a lot of troubles to the readers who want to look up the references.

REPORT

1. You have to submit a progress report at the end of the first semester and a final year project report at the end of the second semester.
2. It is a good idea to write things bit by bit during the year and cut and paste them to the report.
3. Make sure that you use MS Word to spell and grammar check before you hand in.
4. Number the sections according to IEEE format. E.g.

I. INTROUCTION

A. Models of Simple Genetic Algorithm (GA)

1) ...

2) ...

B. ...

II. ...

DO NOT COMMIT PLAGIARISM

1. Everything you write must be original. No cut and paste is accepted, even if you acknowledge the source.
2. The definition is: No two sentences should be the same. It is ok if unintentionally one sentence is the same. It is usually unacceptable when two or more consecutive sentences are the same, though of course it should be judged case by case.
3. If a figure is not drawn by yourself, you should acknowledge the source in the legend. [When you publish a book, you need to ask the author of the figure for permission to reprint the figure in your book.]
4. Before your report is submitted, you need to submit it to an anti-plagiarism system for plagiarism check. Plagiarism will lead to failure or lowering of grades.
5. If a survey is done jointly by two students, then in the report, you should acknowledge at the beginning of the survey that it is written by two students jointly.

MISCELLANEOUS ADVICE

1. When you save a paper, please use a short meaningful filename to help retrieval later: For example, one can save the paper titled “Evolving a neural network location evaluator to play Ms. Pac-man” with the short but informative filename “Evolving NN evaluator to play Ms Pacman”.