CHAPTER 8

CONCLUSION

IN THIS CHAPTER

8.1. Further Scope

8.2. System Testing & Maintenance

8.3. Enhancement Scope

8.4. Conclusion

8. 1. FURTHER SCOPE

As many large projects have some limitations our project CSICAISA has some limitations also. As it a educational project, the project might not meet all the requirements of a real life application. But it could be further developed to increase its efficiency and variations.

8. 2. SYSTEM TESTING & MAINTENANCE

As we all of known that software testing according to industry cannot be properly done in the student's projects. But in spite of that factor we try to maintain the testing methodology in our project according to the software engineering aspects.

The entire project has been tested successfully by us and our faculty member acting upon all the software engineering testing methodology.

Maintenance is such type of activity which much depends on the customer feedback. As our project is CUI (Character User Interface).If there are any difficulties or problems found in further in our project, our team would be fully responsible and bound to be maintain the software at any time of customer reference.

8. 3. ENHANCEMENT SCOPE

The project might not meet all the requirements of a real life application. But it could be further developed to increase its efficiency and variations. There is a scope to develop our project in future and increase its efficiency.

In our project Digit sort is proposed. It has complexity more than some earlier proposed algorithms, but we try to decrease its complexity. It has a special situation when its complexity will decrease with other sorting algorithm that is we try to focus in our project.

Further we try to develop a sorting algorithm that is better or effective as per as complexity concern and maintain the digit sort also.

8. 4. CONCLUSION

By analyzing an algorithm, we mean to study the performance of an algorithm including the assertion of its correctness and a determination of the cost of its execution. Although a given algorithm is often analyzed in a particular way that is most suitable for such an algorithm, we are more interested in general procedures and techniques that can be used to study the performance of classes of algorithms. To be able to talk about general analysis techniques will not only add to our understanding of the behavior of a class of algorithms but will also, in many cases, lead to useful synthesis procedures. A good example illustrating these points is the various techniques that can be used to analyze a class of sorting algorithms which can be modeled as networks made up of comparator modules. In this paper, we discuss several approaches to such an analysis problem. Moreover, synthesis procedures suggested by these analysis techniques will also be presented.