1. CS 5329 Algorithm Design and Analysis
   Given a constant fraction $p$ ( $p = 0.2$ ), and assuming we always have $p$ and $(1 - p)$ proportional partitions for the Quick-Sort algorithm.
   Prove that the total running time for such Quick-Sort is still $\Theta(n \log n)$.
   (Hint: you may use recursion tree and some log algebra.)

2. CS 5391 Survey of Software Engineering
   Describe the following process models and their application scope.
   (a) incremental model
   (b) prototyping model

3. CS 5318 Design of Programming Languages
   Given the following BNF:
   $$
   \begin{align*}
   <expr> &::= (<list>) | a \\
   <list> &::= <list>, <expr> | <expr>
   \end{align*}
   $$
   (a) Draw the parse tree for $((a,a),a,(a))$.
   (b) Design appropriate attribute(s) and write an attribute grammar to compute the level of parentheses. For example, the level of parentheses of the sentence in (a) is 2.
   (c) Decorate the parse tree in (a) with your attributes defined in (b) to show how the level of parentheses is computed.
   (d) Write EBNF rules and/or syntax diagrams for the language.
   (e) Write a recursive-descent parser for the language based on your EBNF rules in (d).

4. CS 5338 Formal Languages
   (a) True or False questions. Example: This is Spring 2012. (True)
      i. The union of a finite number of countably infinite sets is countable. ( )
      ii. The union of a countably infinite number of countably infinite sets is countable. ( )
      iii. Any DFSM halts in at most $|w|$ steps. ( )
      iv. Any NDFSM halts in at most $|w|$ steps. ( )
      v. Any DPDA always halts in at most $|w|$ steps. ( )
      vi. Any NDPDA always halts in at most $|w|$ steps. ( )
      vii. Church-Turing thesis implies that we need computation models that are more powerful than Turing machines for those problems in $\neg SD$. ( )
      viii. Let $\text{ndtime}(f(n))$ be the set of languages that can be decided by some nondeterministic Turing machine in time $O(f(n))$. Every language in $\text{ndtime}(2^n)$ is decidable. ( )
      ix. $NP$-hard is a subset of $NP$-complete. ( )
      x. If $P = NP$, then $NP$-complete languages will be in $P$ as well. ( )
   (b) Short questions.
i. Given the following FSM M, write down three strings that are in $L(M)$.

![FSM Diagram]

ii. Given the following PDA M, write down three strings that are in $L(M)$.

![PDA Diagram]