1. (A) Please define big-Oh notation (i.e. $O(f(x))$), and big-Theta notation (i.e. $\Theta(f(X))$)

(B) Prove or disprove the following equalities.

\[
\sum_{i=1}^{n} i = O(n^3)
\]

\[
\sum_{i=1}^{n} i^2 \log(i) = \Theta(n^3)
\]

2. (10%) Suppose that we have numbers between 1 and 1000 in a binary search tree and want to search for the number 363. Which of the following sequences could not be the sequence of nodes examined and why?

A. 925, 204, 910, 240, 913, 245, 363
B. 903, 230, 911, 244, 898, 258, 359, 363
C. 10, 252, 400, 398, 340, 345, 397, 363
D. 835, 278, 347, 521, 299, 392, 358, 363
E. 102, 399, 387, 219, 266, 382, 381, 278, 363

3. (7%) Please define the complexity classes NP-complete and NP-hard. Are they equal?

There are many records in a file. Now we want to search a record from that file.

Please answer the following questions.

4. For a heap file, assume the unordered records are stored in b blocks. If it involves a linear search, how many blocks do we need to access on average? (4 %)

5. For a sequential file, assume the records are stored in b blocks according to one ordering field. If it involves a binary search, how many blocks do we need to access on average? (4 %)

6. For a hash file, assume the records are stored in b blocks according to a well-designed hash field. If it involves a hash function based search, how many blocks do we need to access in the best case? (4 %)

7. For a file with a primary index access structure, give an example for describing how it works. (7 %)

8. For a file with a clustering index access structure, give an example for describing how it works. (7 %)

9. For a file with a secondary index access structure, give an example for describing how it works. (7 %)
10. (a) Write a program that can construct a binary tree from the input file of “data.dat”. (18%)

(b) Trace your program of constructing the binary tree by using the following set of data in the file. (10%)

"25, 14, 37, 7, 19, 35, 27, 38, 40".

(c) Analyze the complexity of your program. (6%)