Guidelines

1. **Files in this project**
2. /school/student.no.idx is the B+ tree file, which includes the information of the built ASLOG index
3. /Index\_TCSolution is the source code of the project
4. /DLL is the encapsulated functions for building ASLOG and querying recursive rules
5. **DLL functions**
6. *Built in the ASLOG index:*

BPTree\_Index\* CreateIndex(char\* rule, int index\_type, char\* relations, char\* attribute\_list, int attribute\_type)

1. *Query a specific person’s ancestors:*

Person\_AncestorsM(person, btree\_index)

1. *Query the common ancestors of several persons:*

Person\_AncestorsM(person\_list, btree\_index)

1. *Query a specific person’s descendants：*

Person\_descendantM(person, btree\_index)

1. *Query the common descendants of several persons*:

Person\_descendantsM(person\_list, btree\_index)

1. *Calculate the transitive closure:*

Transitive\_closureM(btree\_index)

1. *Calculate the increments of transitive closure:*

IncrementM(btree\_index, New\_tuples)

All of these listed functions store the result **in the memory**.

To store the results **on the disk**, please change the capitalized “M” in the method name to a capitalized “D”. For example, function “Person\_AncestorsD(person, btree\_index)” will store the result on disk when query a specific person’s ancestors.

1. **How to use**
2. To query the ancestor of a specific person:

Invoked function:



The result will be written in the following file



1. To query the common ancestor of two person:

Invoked function:



The result will be written in the following file

