

# Binary Files

4-15-2011

# Opening Discussion

- Finishing the binary heap.

# Motivation

- We have done most of our file work with text files. We saw some binary with streams.
- Binary can be far more efficient for space and speed.
- When combined with direct access, they can let you use disk like nearly unlimited RAM.

# Binary Files

- Binary files mirror the format of data in memory in the file, unlike text.
- Java does some conversions so that numbers are written in a machine independent format.

# Direct/Random Access Files

- Instead of using a stream, it is also possible to have a file where you jump around in it. This only works if you know where to jump to.
- Use `java.io.RandomAccessFile`.
- For links between data you store `Ints` instead of normal references.

# Fixed Record Lengths

- One way to know where things are in the file is to make every entry in the file the same length.
- This makes the file very much like an array of case classes. The Int values are just indexes into the array.
- The only challenge here is picking a record length. This is harder in Java/Scala than in C/C++, which are closer to the machine.

# Variable Record Lengths w/ Index

- If you can't make every record the same length, you make records and indexes. Can use two files, or separate parts of one file.
- Indexes have fixed length and simply *point* to locations in the normal data.
- Harder to update data as length can change. Need to rebuild every so often.

# Code

- In the time we have left we should start writing a little code to do this.



# Minute Essay

- Questions?
- What do you feel absolutely needs to be added to available learning materials for next semester in this course?