Hill Climber and Simulated Annealing Exercises

This is the first exercises in the course. If you do not have the time to do everything, don't worry, the course has just started. Furthermore, exercise session contains two sets of code: Hill Climbing code and Simulated Annealing code. As mentioned in the lecture, the structure of the code is almost identical for the two algorithms, but they should be downloaded separately for each algorithm, i.e. make two different directories, one for the Hill Climber files (in the HC.zip file) and one for the Simulated Annealing files (in the SA.zip files).

Start by working on the Hill Climber files. The same experiments should be done with the Simulated Annealing files.

Download the file:

`HC.zip`

and the set of problem data files:

`test_sets.zip`

. from the homepage. Unpack the files:

- `unzip HC.zip`
- `unzip test_sets.zip`
- Compile the codes by writing: (ignore the warnings for the DataObject.java compilation)
  - `javac DataObject.java`
  - `javac HillClimber.java`
  - `javac MainHC.java`
- Try to run the code a couple of times by writing:

  `java  MainHC`

  . What is the best value you can achieve ? (remember we are maximizing)

- Which of the datasets are you actually using (look in the file MainHC.java) ?
- Change to another dataset (change the name in the file and compile again). Run the code a couple of times, what is the best result you can get ?
- What is the running time of the algorithm ? Take a look in the MainHC.java file and change the running time to a minute. Recompile and run the program again. Can you get better results ?
- Download the large dataset

  `large_testset.zip`
- Test the algorithm on the large dataset again.
- Open the file HillClimber.java. Take a look at the code, and find out what the neighborhood of the algorithm is. Take a look at the line: "Data.DeepClone(Data.Tables, ...". This line is by far the time consuming line, can you devise a way to avoid having to clone the entire solution again ? You **DONT** have to program it.
- Download the Simulated Annealing algorithm and repeat the above exercises, with the Simulated Annealing program.