CS 306	Algorithms and Complexity

Goodness

Iterative Improvement *Exploration*

Goal

The purpose of this exploration is for you to investigate and implement an important iterative improvement algorithm.

Requirements

Using the supplied article by Shawn Carlson as a springboard, do some good, collaborative research on *simulated annealing*, and then implement this iterative improvement algorithm. Apply your implementation to one specific search task: Validate the choices described by the following excerpts from the documentation and source code of `java.util.HashMap`.

```
 * This implementation provides constant-time performance for the basic
 * operations (get and put), assuming the hash function
 * disperses the elements properly among the buckets.

/**
  * Returns a hash value for the specified object.  In addition to
  * the object's own hashCode, this method applies a "supplemental
  * hash function," which defends against poor quality hash functions.
  * This is critical because HashMap uses power-of-two length
  * hash tables.
  *
  * The shift distances in this function were chosen as the result
  * of an automated search over the entire four-dimensional search space.
  */
 static int hash(Object x)
 {
     int h = x.hashCode();

     h += ~(h << 9);
     h ^=  (h >>> 14);
     h +=  (h << 4);
     h ^=  (h >>> 10);

     return h;
 }

 /**
  * Returns index for hash code h.
  */
 static int indexFor(int h, int length)
 {
    return h & (length-1);
 }
```

Use C++ as your implementation language. Note that C++ does not have a '>>>' operator. Note too that Java does not support unsigned types!

Grading Criteria

The breakdown below is meant to guide you in your quest for goodness and success.

- **Collaboration** (15 points)

  Collaborated with at least three other classmates, who are identified.

- **Engagement** (25 points)

  Provided ample evidence of a curious mind at work.

- **Documentation** (30 points)

  Chronicled in some detail the history of decisions made and actions taken, via an activity log in the code — which is also well documented.

- **Correctness of implementation** (30 points)

  Implemented a simulated annealing algorithm, not some other one.

Comments

There are no comments from former students about this exploration. As this semester marks its inception, there are no former students — you are this exploration's intrepid pioneers!