Topics for Projects and Theses

1. **MQP, and graduate work, at MSc or PhD level: Finding average-optimal methods to locate the median of a small set of distinct numbers**

   Several algorithms need, as a subtask, to find the median of a small sets of numbers (5, 7, 9) many times. Hence the interest in doing the selection optimally. Knuth gives optimal costs for 5 and 7 numbers, claiming they are the best possible, but no algorithm or reference (he says they were found by one Mr. Hoey who conducted an exhaustive search).

   During the Fall of 2010 I discovered the way to compute these methods, in principle, and have generated several optimal algorithms (the program prints them out as C-language functions). Several challenges remain. . . .

   (i) The program I wrote was revamped many times, with the data structures reshaped to fit new needs, as research progressed, chasing a moving target. As a result it needs significant reorganization. (MQP).
   (ii) The calculations are extremely heavy. While I expect both algorithmic improvements and computing enhancements are possible, a relatively low-hanging fruit is parallelization. (MQP)

   I expect the following two to be more suitable for graduate work:

   (iii) The method I use to find the optimal policies can probably be vastly simplified, by using a restricted search-space, but more work is needed to design this approach.
   (iv) My discovery uses a Markov decision process, a well-known tool, and there is interest to derive properties of the optimal policies from the recurrence. This is probably the most ambitious part.

   MQP Prerequisites: Nothing beyond CS2022 (discrete mathematics) and the software skills all seniors are expected to have. The project will require careful design, and possibly complicated data structures. Number of students: two or more.

2. **MQP: Experimental investigation of heap-sort.**

   The method of sorting numbers in main storage known as heap-sort has been in a constant state of reinvention, with the objective of surpassing quicksort, which performs marvelously on the average, but has unfortunate worst case behavior. Several variants have been proposed that seem promising but are hard to analyze conclusively, and there is the need for a considerable experimental investigation, with the ultimate goal of tying the observations to suggestions that partial analysis has provided.

   Desirable number of students: two, though it is possible there is room here for three.

   Prerequisites: for CS seniors, probably nothing special, beyond a sufficient interest in the theory of algorithmics and some statistical insight (that might be acquired "on the job.")

---

3. **MQP or graduate work: asymptotic behavior of solutions of divide-and-conquer equations.**

   A divide-and-conquer equation arises when we analyze a thus-called algorithm. A simple equation is as follows:

$$A(n) = 2A(\lfloor n/2 \rfloor) + n + 2, \quad n \geq 2, \quad A(1) = 1.$$

   Over the decade several treatments of such equations have appeared in the literature. The reported results appear at first sight inconsistent.

   This is probably not the case, but a clarification of the subject is desirable, and there is probably some room for improvement as well. A successful project is likely to lead to a journal or conference publication.

   Number of students: one or, preferably, two.

   Prerequisites: This is a mathematical work, but it requires little mathematics more advanced than the discrete math you have taken. It does require a mathematical turn of mind.
   A possible plus is getting to learn to use symbolic calculations, using Maple or Mathematica, since I expect this approach to be suitable for the kind of calculations this project needs.

4. **PhD suggestion: Implementing in the Maple or Sage System the Karr summation algorithms**

   Summation is harder than integration. It is a fact that seems surprising until you get into the heart of the matter. Now there is enough understanding to make it possible, for a very large class of summands, to be able to decide whether a closed-form results exists or not, and for most instances of the first case, derive it.

   In spite of the title, this is not simply an implementation of a completely specified algorithm, since the papers of Michael Karr and several others, while algorithmic in nature, concentrate on the mathematical underpinning. Some interesting research would be required. A similar effort has been carried out in the context of Mathematica, and reading the relevant materials reveals that there is much scope for invention yet, probably of fundamental nature as well.

   The completed work, if successful, stands to be popular tool among CS people, combinatorialists and numerous other groups.