Introduction to Algorithms: 6.006
Massachusetts Institute of Technology — Thursday, October 16
Instructors: Mahdi Cheraghchi, Silvio Micali and Vinod Vaikuntanathan — Problem Set 4

Problem Set 4

All parts are due Thursday, October 30 at 11:59PM. Please download the .zip archive for this problem set, and refer to the README.TXT file for instructions on preparing your solutions. Remember, your goal is to communicate. Full credit will be given only to a correct solution which is described clearly. Convoluted and obtuse descriptions might receive low marks, even when they are correct. Also, aim for concise solutions, as it will save you time spent on write-ups, and also help you conceptualize the key idea of the problem.

Part A

Problem 4-1. [20 points] **Search for pairs**

Give the most efficient algorithm you can in terms of time and space for solving the following problem:

Given a list of n integers in $\{1, \ldots, p-1\}$, where p is a prime in the range $[n^3, 2n^3]$, find the number of pairs (a, b) in the list such that

$$a \cdot b \equiv 7 \mod p.$$

If you use hashing in your algorithm, you are allowed to simplify the analysis assuming simple uniform hashing, and assume that hashing an element takes constant time.

To receive full credit, your algorithm should run in $O(n)$ time and space.

Problem 4-2. [20 points] **How many probes?**

Suppose we use hashing with open addressing to store n distinct items in a hash table with $2n$ entries.

Recall that in the open addressing model, a hash function h_1 is chosen to hash an item k to the table cell at position $h_1(k)$. If the resulting cell is not empty, a second hash function h_2 is used to find an alternative spot $h_2(k)$ for the item, and so on, until some $h_i(k)$ points to an empty cell where the item will be placed. For simplicity, we will assume uniform hashing, that is, assume that all the random variables $h_i(k)$ are uniformly distributed over $\{1, \ldots, 2n\}$ and are independent.

Prove each of the following properties about the above model. In proving each property, you may take the previous ones for granted.

(a) The probability that an insertion takes more than m probes to find an empty slot is at most 2^{-m}.

(b) The probability that an insertion takes more than $2\log_2 n$ probes is at most $1/n^2$.

(c) The probability that one or more of the insertions take more than $2\log_2 n$ probes is at most $1/n$.

(d) Let X be the length of the longest probe sequence for the n insertions. We have $\mathbb{E}[X] = O(\log n)$.

Problem 4-3. [20 points] **Algorithms workshop**

MIT is organizing a workshop on algorithms and needs your help to design the workshop's program. There are n participants registered for the workshop, and the workshop consists of m tracks, each about a particular topic. Each participant has chosen one or two tracks to attend (so we have $m \leq 2n$). The organizers would like to know whether they can organize the workshop to be held over a weekend (Saturday and Sunday). Each track takes a whole day, so the program should be designed so that the participants do not have to attend two different tracks on the same day. Design an algorithm that runs in $O(n)$ time and decides whether it is possible to organize the workshop with the required constraints. The input to the algorithm is n list, where the ith list (of size 1 or 2) specifies the tracks that the ith participant has registered for.

Hint: You may find it useful to phrase the problem in a graph-theoretic sense, and then use an algorithm inspired by breadth first search.

Part B

Problem 4-4. [40 points] **So many humans, such little Na'vi**

James Cameron has received much help from the 6.006 students this semester, so he comes back to MIT for help with his next problem. In *Avatar 2* Cameron wants more humans to be able to operate the Na'vi-human hybrids, technically called avatars. As any avid fan – one who bleeds blue for Avatar – would know, the process of making avatars is extremely expensive so Cameron is constrained in his numbers. Moreover, avatars are engineered to genetically match a human counterpart, and thus each avatar is operable by only the human they were created for (or an identical twin). Cameron builds into the plot of *Avatar 2* a new research breakthrough coming from the MIT Whitehead Institute where researchers have found a way to pair humans and avatars that are not genetically identical. There still needs to be enough genetic similarity for a connection to be made, however. The scientists in the movie work with the MIT researchers to come up with a way to find Na'vi and human DNA similarity, but their process takes too long when running comparisons on 3 - 6 billion base pairs per DNA sequence. They now consult algorithm students for help calculating DNA similarities.

(a) Implement the function `dna_match(navi_dna, human_dna)` that takes in a Na'vi DNA sequence of length n and a human DNA sequence of length m, where $m = O(n)$, and returns the nucleotide sequence, s, with maximum length, such that s exists in both the Na'vi and human DNA. Note that while human DNA has only 4 nucleotides (A,T,C,G), Na'vi DNA is composed of 26 nucleotides represented by all of the letters in the english alphabet.

Your solution should run in $O(n \log(n))$ time. **Hint**: You may find the rolling hashing technique of Rabin-Karp's string search algorithm an inspiration for this problem. You are free to use any python standard functions and libraries. Aside: Humans have 23 chromosomes and it just so happens that 23 is a prime number. Coincidence? Probably, but interesting none-the-less.

Part C

Problem 4-5. [5 bonus points] **Piazza poll**

Please fill out the Piazza poll indicating how much time you have spent on each part of this problem set. Indicate in your write up whether or not you complete the poll.