6.856 — Randomized Algorithms

David Karger

Handout #11, October 9, 2000 — Homework 6, Due 10/16

M. R. refers to this text:
Motwani, Rajeez, and Prabhakar Raghavan. *Randomized Algorithms*. Cambridge: Cambridge University Press, 1995.

1. Show that there is no perfect hash family mapping from m to n of size polynomial in m if $2n \leq m \leq 2^{o(n)}$.

2. MR 8.20. Consider the problem of deciding whether two integer *multisets* (that is, sets with repeated elements allowed) are identical in the sense that each integer occurs the same number of times in both sets. The problem can be solved by sorting the two sets. Give an efficient randomized algorithm for comparing two multisets using universal hash functions. Compare with the sorting-based solution. Discuss the relative merits of the two algorithms, considering issues such as the model of computation (eg unit cost RAM, where a multiply takes constant time, vs. log-cost RAM, where a b-bit multiply takes $O(b \log b)$ time), the size of the integers being operated upon, and ways the algorithm can fail.

3. MR 7.4. Consider the same multiset comparison as the previous problem. Suggest a way to represent this as a problem involving the verification of a polynomial identity, and thereby obtain an efficient randomized algorithm. Compare its relative merits with those of the two algorithms of the previous problem.

4. In class, we discussed how fingerprinting can be used to identify all occurrences of a pattern in a text string. Show how to preprocess the text string so that we can quickly test for the presence of an input pattern. Show that:

   (a) For any particular pattern size, after expected linear time (in the text size) preprocessing (and use of at most linear space), we can answer any query as to the number of occurrences of a given pattern in expected time proportional to the pattern size. Our answer should be correct with high probability.

   (b) After polynomial expected time preprocessing, we can output all locations where a given pattern appears in worst case time proportional to the pattern size and number of appearances. Our answer should always be correct.

   (c) (Optional) Figure out how to perform the preprocessing of the previous part in deterministic polynomial time. **Hint:** Show that from any set of n^4 primes, you can find one prime p such that all the patterns are different mod p. Now distinguish the residues.

5. *(optional)* MR 8.28. Improve on the constant in the space bound for perfect hashing. Our scheme in class uses $13s$ space for s items. In fact one can get by with $s + o(s)$ space. **Hint:** increase the size of the primary hash table and observe that most of the

bins will be empty. Find an efficient scheme (by packing several “short ints” into a single $\log m$ bit word) for packing together the non-empty bins while creating secondary hash tables only for the bins of size greater than 1.