Searching With Just One Probe
Exposition by William Gasarch (gasarch@cs.umd.edu)

1 Introduction

Definition 1.1 The *Cell Probe Model* for search is as follows:

1. The size of the universe is U. The universe is $\{1, \ldots, U\}$.

2. The number of elements from the universe that we will store is n.

3. The function PUT takes $A \in \binom{[U]}{n}$ and outputs the elements of A in some order. This tells us how to store A in an array.

4. An algorithm $FIND$ that, on input $x \in U$, probes the array (asks 'What is in cell c'), and based on the answer probes another cell, etc, and then says either x is in A, or x is not in A.

In class we showed, using Ramsey Theory, the following:

Theorem 1.2 *If* $U \ge R(n, 2n-1, n!)$ *(n-ary Ramsey,* $2n-1$ *sized homog set,* $n!$ *colors) then no cell probe algorithms can do better than* $\lceil (\rceil \log(n+1))$ *probes. (Hence sorting is optimal.)*

We will now look into what you can with just ONE probe.

2 If $U = 2n-2$ Then There is a 1-Probe Algorithm

We first do an example. Let $n = 5$ and $m = 8$. We think of the 5-sized-array as being five houses. We use the term House and Cell to mean the same thing.

- Both 1 and 6 want to live in house 1. 6 is called *the upper tenant.* 1 is called *the lower tenant.*

- Both 2 and 7 want to live in house 2. 7 is called *the upper tenant.* 2 is called *the lower tenant.*

- Both 3 and 8 want to live in house 3. 8 is called *the upper tenant.* 3 is called *the lower tenant.*

- 4 wants to live in house 4. 4 is called *the lower tenant.*

- 5 wants to live in house 5. 5 is called *the lower tenant.*

Given a set A of 5 elements from $\{1, \ldots, 8\}$ we want to order them and put them in the houses such that the following happens:

- Let $1 \le i \le 3$. If only one tenant who wants to live in house i is in A, then that tenant is stored house i.

- Let $1 \le i \le 5$. If none of the tenants who want to live in house i are in A then some other houses upper tenant is stored in house i.

- Let $1 \leq i \leq 3$. If both tenants want to live there than some other houses lower tenant is stored in house i.

IF we can pull this off THEN the 1-probe algorithm is as follows:

FIND

1. Input x where $1 \leq x \leq 8$
2. Let t be the house where x is a potential tenant.
3. Probe(t).
4. If House t has x, then output YES.
5. If House t has the other tenant than output NO.
6. If House t has an upper tenant then output NO.
7. If House t has a lower tenant then output YES.

NOW we have to say how we accomplish storing items so that the needed conditions happen.

PUT

1. Input $A \in \binom{[8]}{5}$.
2. If $(1 \in A) \oplus (6 \in A)$ then put whichever one is in A into house 1.
3. If $(2 \in A) \oplus (7 \in A)$ then put whichever one is in A into house 2.
4. If $(3 \in A) \oplus (8 \in A)$ then put whichever one is in A into house 3.
5. Put every upper tenant that is left into a house which nobody wanted to be in.
6. Take the lower tenants that are left. Put each one in any empty house, but make sure its NOT the house that it is a lower tenant of.

Example: $A = \{1, 2, 5, 7, 8\}$.

From the first three steps we put 1 in House 1 (since $1 \in A$ but $6 \notin A$), and 8 into House 3 (since $3 \notin A$). So far we have

1	2	3	4	5
1		8		

All that are left is $2, 5, 8$.

The only house that nobody wants to live in is house 4, so put upper-tenant-7 there. So we have:

1	2	3	4	5
1		8	7	

Now whats left is $2, 5$ both of which are lower-tenants. Since you don't want them to go to their own houses, swap them:

1	2	3	4	5
1	5	8	7	2