22C:44 Homework 6 Solution

1. Problem 12.3-4: 61 hashes to 700, 62 to 318, 63 to 936, 64 to 554, and 65 to 172.

   Look in the practice problem solutions for problem 12-4.1.

2. To determine when the probe sequence $h(k,0), h(k,1), \ldots, h(k,m-1)$ is a permutation consider $h(k,i) - h(k,j)$ for $0 \leq j < i \leq m-1$. We see that

   $$h(k,i) - h(k,j) = c(i-j) \bmod m.$$

   Note that $i-j$ can take on any value between 1 and $m-1$. Now suppose that c and m have a common factor $f > 1$. Note that $m/f \leq m-1$. Let $c = f \cdot d$. Then $c(i-j)$ can be rewritten as $df(i-j)$ and letting $(i-j)$ take on the value m/f we get that $c(i-j) = df \cdot m/f = d \cdot m$. This implies that $h(k,i) - h(k,j) = 0$. So what we have shown is that if c and m have a common factor greater than 1 then the probe sequence is not a permutation. On the other hand, if c and m have no common factors greater than 1 then the probe sequence is a permutation.

   No, this hash function is no better than linear probing. Primary clustering is a problem here as well. Long contiguous blocks of filled slots is not the problem now. Now the problem is long filled sequences of slots in which each slot is c slots away from the previous slot.

3. Look in the practice problem solutions for this.

4. Let $C = \{c_1, c_2, \ldots, c_M\}$ be the greedy solution and let $F = \{f_1, f_2, \ldots, f_N\}$ be an optimal solution. We assume that $N < M$ and derive a contradiction. Without loss of generality assume that C and F are both ordered in non-increasing order. Let i be the smallest integer in the range $\{1, 2, \ldots, M\}$ such that $c_i \neq f_i$. In other words, $c_1 = f_1, c_2 = f_2, \ldots, c_{i-1} = f_{i-1}$. Since C and F are change for the same value we have that $\sum_{j=1}^{M} c_j = \sum_{j=1}^{N} f_j$. This also implies that

   $$\sum_{j=i}^{M} c_j = \sum_{j=i}^{N} f_j.$$

   Furthermore, because c_j's are chosen greedily, we have that $c_i > f_i$. In particular, suppose that $c_i = 2^s$ and $f_i = 2^p$ for some $p < s$. Also note that if $f_j = f_{j+1}$ for some $j \geq i$ then f_j and f_{j+1} can be replaced by $2 \cdot f_j$, which is also a power of 2 that is no greater than 2^s. So we can assume that elements in $\{f_i, f_{i+1}, \ldots, f_N\}$ are all distinct. Since $f_i = 2^p$, we have that

   $$f_{i+1} \leq 2^{p-1}, f_{i+2} \leq 2^{p-2}, \ldots$$

   and so the sum

   $$\sum_{j=i}^{N} f_j \leq 2^p + 2^{p-1} + \cdots = 2^{p+1} - 1 \leq 2^s - 1 < 2^s = c_i.$$

   This contradicts the fact that $\sum_{j=i}^{M} c_j = \sum_{j=i}^{N} f_j$.

5. Let us call the greedy algorithm discussed in class (and described in the book) the "correct greedy algorithm" since we have shown the correctness of this algorithm. Let us call the algorithm described in the homework assignment, simply the "greedy" algorithm. We show that the greedy algorithm is correct by showing that the solution it produces is equal in size to the solution produced by the correct greedy algorithm. To show this we need to explore the structure of the solution produced by the correct greedy algorithm some more.

   Let $A = \{a_1, a_2, \ldots, a_n\}$ be the given set of intervals and let the solution produced by the correct greedy algorithm be $G = \{a_{i_1}, a_{i_2}, \ldots, a_{i_k}\}$. Recall that we use the notation l_i and r_i to denote the left endpoint and the right endpoint respectively of the interval a_i. Partition the set of all intervals A into groups $G_1, G_2, \ldots, G_k$ with G_j defined as the set of intervals

$$G_j = \{a_i \mid r_{i_{j-1}} < l_i < r_{i_j} < r_i\}.$$

   In other words, G_j is the set of intervals that are compatible with $a_{i_1}, a_{i_2}, \ldots, a_{i_{j-1}}$ and incompatible with a_{i_j}. Another way to think about G_j is that it is the set of the intervals removed from A in the jth step of the correct greedy algorithm.

   We will now show that the greedy algorithm picks exactly one interval from each set G_j, thereby picking exactly k intervals. We show this by induction. Let A_s be the set of intervals remaining in consideration after s steps of the greedy algorithm. Note that this means that $A_0 = A$. Define a set J_s as follows

$$J_s = \{j \mid G_j \cap A_s \neq \emptyset\}.$$

   Note that J_s denotes the G_j's that contribute elements to A_s. The induction hypothesis is that after s steps of the greedy algorithm J_s has size $k - s$. This means that after k steps, $J_s = \emptyset$ and therefore $A_s = \emptyset$. In other words, what is happening is that in each step of the greedy algorithm we pick an interval from some set G_j to be in our solution and as a result whatever remains of the set G_j in A gets removed; also as a result a few but not all intervals, from other sets $G_{j'}$ may be removed from A.

   The base case is when $s = 0$. Clearly, $J_0 = \{1, 2, \ldots, k\}$ and therefore the size of J_0 is k.

   Let the inductive hypothesis hold for some s. We will show that it is also true for $(s + 1)$. Suppose that in step $(s + 1)$ the greedy algorithm picks an interval a_i in G_j to be placed in the solution. Note that intervals in G_j are all mutually incompatible and therefore when intervals that are mutually incompatible with a_i are deleted from A, every interval in G_j is removed. This means that $j \in J_s - J_{s+1}$. For any interval a, let $degree(a)$ represent the number of intervals it is incompatible with. Let p be the largest value in J_s smaller than j. a_i could be incompatible with some intervals in G_p, but if it is incompatible with every interval in G_p, then we can show that $degree(a_i) > degree(a_{i_j})$ and we would have picked a_{i_j}. Similarly, we can show that if q is the smallest value in J_s larger than j then a_i could be incompatible with some intervals in G_q, but not all. Therefore, $J_s - J_{s+1} = \{j\}$ and therefore the size of J_{s+1} is $k - (s + 1)$.