

Homework 4

Due: 5:00pm, March 27, 2009 (Late deadline: 5:00pm, March 30, 2009)

Total Points: 40

You may submit your solution via email to holder@wsu.edu (preferred), or you may submit hardcopy in class or to my office (EME 227) by the above deadline. If you submit via email, please use PDF or other common electronic format.

Note: Your solutions should match the result from using the code in the textbook, which may be different from the results obtained by applets or other implementations.

1. (20 points) For each of the different hash tables described below, show the final hash table after inserting the keys 42, 33, 45, 5, 14, 58, 84, 6, 2, 40 (in this order) into an initially empty table.
 - a. (5 points) A hash table of size $M=7$ using collision-resolution by chaining and the hash function $hash(x) = x \bmod M$. Your table should look similar to the one in Figure 5.5.
 - b. (5 points) A hash table of size $M=11$ using collision-resolution by open-addressing and the linear probing hash function $h_i(x) = (hash(x) + f(i)) \bmod M$, where $hash(x) = x \bmod M$, and $f(i) = i$. Your table should look similar to the one in Figure 5.19.
 - c. (5 points) A hash table of size $M=11$ using collision-resolution by open-addressing and the quadratic probing hash function $h_i(x) = (hash(x) + f(i)) \bmod M$, where $hash(x) = x \bmod M$, and $f(i) = i^2$. Your table should look similar to the one in Figure 5.19.
 - d. (5 points) A hash table of size $M=11$ using collision-resolution by open-addressing and the double hashing function $h_i(x) = (hash(x) + f(i)) \bmod M$, where $hash(x) = x \bmod M$, and $f(i) = i * hash_2(x)$, and $hash_2(x) = R - (x \bmod R)$, where $R = 7$. Your table should look similar to the one in Figure 5.19.
2. (5 points) Show the final binary heap after performing the linear-time buildHeap algorithm on the sequence 42, 33, 45, 5, 14, 58, 84, 6, 2, 40. Your final binary heap should be drawn as a tree.
3. (3 points) Show the binary heap after performing a deleteMin on the binary heap on the right side of Figure 6.11. Your final binary heap should be drawn as a tree.
4. (5 points) Show the final binomial heap after inserting the keys 42, 33, 45, 5, 14, 58, 84, 6, 2, 40 (in this order) into an initially empty binomial heap. Your final binomial heap should be drawn as a forest of trees.

5. (3 points) Show the binomial heap after performing a deleteMin on the binomial heap in Figure 6.49. Your final binomial heap should be drawn as a forest of trees.
6. (4 points) Show the binomial heap after merging the following binomial heap with the binomial heap at the bottom of Figure 6.60. Your final binomial heap should be drawn as a forest of trees.

