B505/I500: Applied Algorithms

HW2 (Due: **Feb. 15 Friday, 5pm**)

https://iu.instructure.com/courses/1771436

1. (10 pts) Illustrate the operation of extracting the maximum element on the binary heap A=<15, 13, 9, 5, 12, 8, 7, 4, 0, 6, 2, 1>;
2. (10 pts) You are given an array of *n* elements, and you notice that some of these elements are duplicates; that is they appear more than one time in the array. Devise an algorithm to remove all duplicates from the array in time O(n log n).
3. (10 pts) A sequence of n operations is performed on a data structure. The ith operation costs i if i is an exact power of 2, and 1 otherwise. Use aggregate analysis to determine the amortized cost per operation.
4. (10 pts) A sequence of stack operations is performed on a stack whose size never exceeds **k**. After every **k** operations, a copy of the entire stack is made for backup purposes. Show that the cost of **n** stack operations, including copying the stack, is **O**(**n**) by assigning suitable amortized costs to the various stack operations.
5. (10 pts) Suppose we wish not only to increment a counter but also to reset it to zero (i.e., make all bits in it 0). Show how to implement a counter as an array of bits so that any sequence of increment or reset operations takes time O(n) on an initially zero counter.
6. (10 pts) What is the total cost of executing n of the stack operations PUSH, POP, and MULTIPOP, assuming that the stack begins with s_0 objects and finishes with s_n objects?
7. (10 pts) Illustrate the insertion of the keys 5, 28, 19, 15, 20, 33, 12, 17, 10 into a hash table with collisions resolved by chaining. Let the table has 9 slots, and let the hash function be h(k) = k mod 9.
8. (10 pts) For the set of {1, 4, 5, 10, 16, 17, 21} of keys, draw binary search trees of height 2, 3, 4, 5, and 6.
9. (10 pts) Suppose that the search for the key *k* in a binary search tree ends up in a leaf. Consider three sets: 1) the keys to the left of the search path; 2) the keys on the search path; 3) the keys to the right of the search path. One claims that any three keys a, b and c from these three sets, respectively (say a from 1, b from 2 and c from 3), it is always true that $a \leq b \leq c$. Is this claim true? Explain your answer.
10. (10 pts) We can sort a given set of n numbers by first building a binary search tree containing these numbers (using Tree_insert algorithm repeatedly to insert the numbers one by one) and then printing the number in an inorder tree walk. What are the worst-case and best-case running times for this sorting algorithm?