CEN213 Data Structures Final Exam

25.01.2021

QUESTIONS

(***There are 3 questions in total, you should submit your solutions until 15:00***)

1. **(30 points)** Assume that you are given a large student list that contains student id number, name, surname, and phone number of 50000 students in our university. It is also given that the student list is stored in an array in ascending order of student id numbers, and too many search operations with respect to student id numbers are done over the student array. Answer the following questions:

   a) If binary search algorithm is applied on the array that stores the whole student list, compute maximum number of key comparisons that is to be done for 1 successful search.

   b) Assume that 70% of the search operations are done for highly popular 1000 students. So, you define an array A to store these 1000 students in sorted order, and store the remaining 49000 students in another array B also in sorted order. Then, you apply binary search on array A, if the student can't be found, then apply binary search on array B. Compute the expected number of key comparisons made for finding 1 student.

   c) Compare the above two methods in terms of number of key comparisons made, and explain which method is better in terms of running time complexity (or number of key comparisons).

2. **(40 points)** In this question, you are required to implement **enqueue** and **dequeue** operations of **a priority queue** that stores integer elements. For this implementation you should use array based min heap where the root has the minimum value having the highest priority, and each node has a value that is less than or equal to its children. **You can use heap functions int left(int i), int right(int i), int parent(int i) (given in the heapsort chapter)** that return the array index of the left child, right child, and parent of node i without writing their source codes. Write C codes for the below questions assuming that **heap_size** is a global variable that stores the index of the last element in the heap, and **PQ** is an array of integer that is used for keeping the priority queue (also the min heap):

   a) Re-write the **int heapify(int i)** function that you were given in the heapsort chapter so that your function makes operations for min heap.

   b) Write a C function **void enqueue(int item)** to insert a new integer item to the priority queue using the min heap structure. Your enqueue function should work as follows:
   - Insert the new element at the end of the heap.
   - Call heapify() for the parent of the new element.
   - Continue to call heapify() for all ancestors of the new element including the root node.

Good Luck…

Then, estimate the running time complexity of your function in big-O notation and explain your answer.

c) Write a C function **int dequeue(void)** to remove and return the item from the priority queue using the min heap structure. Your dequeue function should work as follows:

- Return the element from the root node.
- Remove the last element from the heap and copy it into the root node.
- Call heapify() for the root node.

Then, estimate the running time complexity of your function in big-O notation and explain your answer.

3. (30 points) In a computer science department, it is required to prepare a schedule for makeup exams. There are 5 students who will take the exams from the 6 courses (A, B, C, D, E, and F). The courses from which each student will take a makeup exam are given below:

Student 1 will take exams from courses B, C, D
Student 2 will take exams from courses A, D, F
Student 3 will take exams from course E
Student 4 will take exams from courses A, E
Student 5 will take exams from courses B, D

a) According to the data given above, by using the graph coloring algorithm, determine the list of courses whose exams can be made at the same time without any conflict for the students. What is the minimum number of exam sessions that must be made without causing any conflict between courses and students? Explain your answer.

b) According to the graph that you have drawn above, give the order of the nodes that are traversed according to the Breath-First Traversal algorithm if the traversal starts from vertex A.

Good Luck…