

بسمه تعالی



دانشگاه صنعتی امیرکبیر

دانشکده مهندسی کامپیوتر

تمرین سری اول درس طراحی الگوریتم‌ها

موعد تحویل : دوشنبه ۱ اردیبهشت ساعت ۲۳:۵۵

توجه:

۱. تمرین های خود را حداکثر تا موعد مقرر در قالب یک فایل PDF به نام HW1_stdNum که stdNum برابر با شماره دانشجویی است بارگزاری کنید.
۲. در صورت نیاز می‌توانید سوالات خود را از طریق رایانامه hosseinpour@aut.ac.ir با تدریس‌یار درس در میان گذارید.

موفق باشید ☺

1. Insertion sort can be expressed as a recursive procedure as follows. In order to sort $A[1 \dots n]$, we recursively sort $A[1 \dots n-1]$ and then insert $A[n]$ into the sorted array $A[1 \dots n-1]$. Write a recurrence for the running time of this recursive version of insertion sort.
2. Describe a $\Theta(n \lg n)$ -time algorithm that, given a set S of n integers and another integer x , determines whether or not there exist two elements in S whose sum is exactly x .
3. Let $f(n)$ and $g(n)$ be asymptotically positive functions. Prove or disprove each of the following conjectures.
 - a. $f(n) = O(g(n))$ implies $g(n) = O(f(n))$.
 - b. $f(n) + g(n) = \Theta(\min(f(n), g(n)))$.
 - c. $f(n) = O(g(n))$ implies $\lg(f(n)) = O(\lg(g(n)))$, where $\lg(g(n)) \geq 1$ and $f(n) \geq 1$ for all sufficiently large n .
 - d. $f(n) = O(g(n))$ implies $2^{f(n)} = O(2^{g(n)})$.
 - e. $f(n) = O((f(n))^2)$.
 - f. $f(n) = O(g(n))$ implies $g(n) = \Omega(f(n))$.
 - g. $f(n) = \Theta(f(n/2))$.
 - h. $f(n) + o(f(n)) = \Theta(f(n))$.
4. Show that the solution to $T(n) = 2T(\lfloor n/2 \rfloor + 17) + n$ is $O(n \lg n)$.
5. Use a recursion tree to determine a good asymptotic upper bound on the recurrence $T(n) = 3T(\lfloor n/2 \rfloor) + n$. Use the substitution method to verify your answer.
6. Use the master method to give tight asymptotic bounds for the following recurrences.
 - a. $T(n) = 4T(n/2) + n$.
 - b. $T(n) = 4T(n/2) + n^2$.
 - c. $T(n) = 4T(n/2) + n^3$.