CS5592 Homework Suggestions Assigned Lecture 2
Try to complete within one or two weeks.

---

1. Review material on Useful Mathematical Facts, that is: exponentials, logarithms, integer functions, summations, limits and L'Hôpital's rule when undefined ratio's come not place.

---

2. Review material on Algorithm analysis (time complexity, and asymptotic behavior of functions), as presented in the hand-outs, and supplemented by e.g. text book. Alternately, use the material handed out that was copied from Johnsonbaugh's book that was (is?) used in CS191 and CS291.

---

3. Test your knowledge by doing as many exercises from this material as need be to become good at it. In particular, do the graduate level exercises copied from Parberry's book.

---

4. If the complexity of an algorithm is $T(n) = c\, n\, \sqrt{n}$, and if it takes 256 μseconds to solve a problem of size $n = 1024$, then how many μseconds does it take to solve a problem of size $n = 4096$?

---

5. If the complexity of an algorithm is $T(n) = c\, n\, \sqrt{n}$, and if it takes 256 μseconds to solve a problem of size $n = 1024$, then what is the maximal problem size that can be solved in 4096 μseconds?

---

6. The complexities of two algorithms is $T_1(n) = 22\, n\, \sqrt{n}$ and $T_2(n) = 3\, n^2$. For which intervals of values of n is the first algorithm faster than the second?

---

7. The Find-algorithm for an Array as presented in class had best, average, and worst case time complexities of $T^B = 1$, $T^A \approx \frac{1}{2}n$, and $T^W = n$. The probability for performing a search for an element is assumed uniform. How do these complexities change if:

1. The elements are first sorted by value, after which a binary search is implemented?
2. The elements are not yet sorted, and inserted one-by-one into a binary search tree. A Binary Search Find is now performed.
3. The elements are first sorted by value, and inserted one-by-one into a binary search tree. A Binary Search Find is now performed.
4. The elements are first sorted by value, after which the middle element becomes the root a of binary search tree, and the binary search tree is recursively constructed. A Binary Search Find is now performed.

---

8. Repeat the above question, but now the probabilities for the elements are not uniform, but could be anything. That is, $p_i \neq \frac{1}{n}$ with $\sum_i p_i = 1$, where p_i is used for the probability that the wanted element is at location i: $p_i = \Pr[A[i] == x]$.

---

9. Repeat the above question, but now the probabilities for the elements are not uniform, but could be anything. That is, $p_i \neq \frac{1}{n}$ with $\sum_i p_i = 1$, where p_i is used for the probability that the wanted element is at location i, $\Pr[A[i] == x]$. Additionally, assume that these probabilities are decreasing: $p_1 \geq p_2$ and so on: $p_{i-1} \geq p_i$.

---

10. Consider additional strategies for the Find-algorithm where now the probabilities for the elements are not uniform, but could be anything. That is, $p_i \neq \frac{1}{n}$ with $\sum_i p_i = 1$, where p_i is used for the probability that the wanted element is at location i, $\Pr[A[i] == x]$.

1. The elements are first sorted by probability, and inserted one-by-one into a binary search tree.

Does this strategy lead to a binary search tree such that the average time complexity for a find is optimal as compared with all other binary search trees? Give an example with 3, 4, or 5 elements that shows that this strategy does not always result in an optimal binary search tree.

---

11. Given that the probabilities for the elements are not uniform, that is, $p_i \neq \frac{1}{n}$ with $\sum_i p_i = 1$, where p_i is used for the probability that the wanted element is at location i, $\Pr[A[i] == x]$. Which strategy would you employ if the objective is to minimize the worst case time complexity?

12. For each function $T(n)$ in the leftmost column, indicate with Y(es) or N(o) when **both** conditions are met. You do not need to justify here, but make sure you are able to reason and justify when asked, e.g. by proving with limits, or by providing examples for the functions involved, or by providing counter examples for the functions involved, or any other method, as long it is appropriately rigorous.

$T(n) =$	$=\mathcal{O}(\lg n)$ $\neq \Theta(\lg n)$	$=\mathcal{O}(n)$ $\neq \Theta(n)$	$=\mathcal{O}(n \ \lg n)$ $\neq \Theta(n \ \lg n)$	$=\mathcal{O}(n^2)$ $\neq \Theta(n^2)$	$=\mathcal{O}(n^2 \lg n)$ $\neq \Theta(n^2 \lg n)$	$=\mathcal{O}(n^3)$ $\neq \Theta(n^3)$	$=\mathcal{O}(2^n)$ $\neq \Theta(2^n)$
$n + 40^{40} + \lg n$							
$40 \ n \log n$							
$n \ (\log n)^2$							
$400n + n\sqrt{n}$							
$2 \ n + 2^n + n^2$							

Use your own imagination, and change the conditions such as "**both** $=\mathcal{O}(n \ \lg n)$ **and** $\neq \Theta(n \ \lg n)$" to other asymptotic classifications.

13. Yes or No. You do not need to justify here, but make sure you are able to reason and justify when asked, e.g. by proving with limits, or by providing examples for the functions involved, or by providing counter examples for the functions involved, or any other method, as long it is appropriately rigorous.

1)	Given that $T = \Theta(n)$. Does this imply that both $T = \mathcal{O}(n)$ and $T = o(n)$?	YES	NO
2)	Given both $T = \mathcal{O}(n)$ and $T = o(n)$ Does this imply that both $T = \mathcal{O}(n^2)$ and $T = o(n^2)$?	YES	NO
3)	Given both $T = \mathcal{O}(n)$ and $T = \Omega(n)$ Does this imply that $T = \Theta(n)$?	YES	NO
4)	Given that $T = \Theta(n)$. Does this imply that $T = \mathcal{O}(n)$ AND $T = \Omega(n)$?	YES	NO
5)	Given that $0 < T(n) < 4 \ n^2$ for all n larger than $N = 30$ Does this imply that $T = \Theta(n)$?	YES	NO
6)	Given that $T = \Theta(n)$. Does this imply that there are constants $c > 0$ and $N > 0$ such that $0 < c.T(n) < n$ for all n larger than N?	YES	NO
7)	Given that $T_1 = \Theta(\ T_2\ )$. Is it possible that $T_2(n) < T_1(n)$ for all $n \leq 1,000$?	YES	NO
8)	Given that $T_1 = \Theta(\ T_2\ )$. Is it possible that $2 \ T_2(n) < T_1(n)$ for all $n \geq 1,000$?	YES	NO
9)	Given that $T = o(\ n^2\ )$. Does this imply necessarily that $T(n) = o(n \ \lg(n)\ )$?	YES	NO
10)	Given that $T_1 = \Theta(\ T_2\ )$, and that $T_2 = \Theta(\ T_3\ )$. Is it always the case that $\lim \frac{T_1(n)}{T_2(n)} \leq \lim \frac{T_2(n)}{T_3(n)}$ as $n \to \infty$?	YES	NO
11)	Given that $T_1 = \mathcal{O}(\ T_2\ )$, and that $T_2 = \mathcal{O}(\ T_3\ )$. Is it always the case that $\lim \frac{T_1(n)+T_2(n)}{T_2(n)+T_3(n)}$ is finite, as $n \to \infty$?	YES	NO
12	Given that $T = \Theta(n)$. Does this imply that $T = \mathcal{O}(n^2)$?	YES	NO
13	Given that $a \in \mathbb{N}$ and $b \in \mathbb{N}$ Does this imply that $(n - a)^b = \Theta(n^b)$	YES	NO
14	Given that $T_1 = \Theta(\ n^2\ )$ *and* $T_2 = \mathcal{O}(\ n^3\ )$. Does this imply that $T_1 + T_2 = \Theta(\ n^3\ )$?	YES	NO
15	Given that $T_1 = \Theta(\ n^2\ )$ *and* $T_2 = \mathcal{O}(\ n^3\ )$. Does this imply that $T_1 + T_2 = \Omega(\ n^2\ )$?	YES	NO
16	Given that $T_1 = \mathcal{O}(\ n^2\ )$ *and* $T_2 = \mathcal{O}(\ n^2\ )$. Does this imply that $T_1 + T_2 = \mathcal{O}(\ n^2\ )$?	YES	NO
17	Given that $T_1 = \mathcal{O}(\ T_3\ )$ *and* $T_3 = \mathcal{O}(\ n^2\ )$. Does this imply that $T_1 + T_3 = \mathcal{O}(\ n^2\ )$?	YES	NO
18	Given that $T_1 = \Theta(\ n^2\ )$ *and* $T_2 = \Theta(\ n^2\ )$. Does this imply that $T_1 \times T_2 = \Theta(\ n^2\ )$?	YES	NO
19	Given that $T_X(n) = \Theta(n \ \lg n)$ and $T_Y(n) = \Theta(n^2)$. Is it possible that $T_Y(1000) < T_X(1000)$?	YES	NO
20	Given that $T_X(n) \sim n \ \lg n$ and $T_Y(n) \sim n^2$. Is it possible that $T_Y(1000) < T_X(1000)$?	YES	NO