Selected Problems in CLRS

Section 8

Notifications

Problem Difficulty (count with star)

1. you can solve w/o the brain
2. you can solve if you think a bit
3. you can solve if you think carefully
4. you might solve if you push yourself
5. you can solve if you use other's brain

Exercise

8.1-1 $\star\star$

What is the smallest possible depth of a leaf in a decision tree for a comparision sort?

> **Note :** Providing lower bound of a problem is very useful, thanks to this we do not have to try to make faster comparison sort algorithm. Similary, NP class problems provide the evidence that given problem can't solve in poly-time (with high probability). NP class will be covered (prof. moon always teaches it).

8.2-1 $\star\star$

Using Figure 8.2 as a model, illustrate the operation of Counting-Sort on the arrya $A = \langle 6, 0, 2, 0, 1, 3, 4, 6, 1, 3, 2 \rangle$.

8.2-1 $\star\star$

Prove that Counting-Sort is stable.

8.3-1 $\star\star$

Using Figure 8.3 as a model, illustrate the operation of Radix-Sort on the following list of English word: COW, DOG, SEA, RUG, ROW, MOB, BOX, TAB, BAR, EAR, TAR, DIG, BIG, TEA, NOW, FOX.

8.3-4 $\star\star\star$

Show how to sort n integers in the range 0 to $n^3 - 1$ in $O(n)$ time.

> **Hint :** exploit radix sort

8.4-1 $\star\star$

Using Figure 8.4 as a model, illustrate the operation of Bucket-Sort on the array $A = \rangle .79, .13, .16, .64, .29, .30, .89, .53, .71, .42 \rangle$.

8.4-2 $\star\star\star$

Explain why the worst-case running time for bucket sort is $\Theta(n^2)$. What simple change to the algorithm preserves its linear average-case running time and makes its worst-case running time $O(n \lg n)$