CS 4349 — Advanced Algorithm Design and Analysis, Homework 3

https://iniyanjoseph.com | iniyanijoseph@gmail.com

Question 1

a

Failing Substitution Proof:
$T(n) = 2T(n-1) + 1$ has the solution $T(n) = O(2^n)$ Assume that $T(n) \leq c \cdot 2^n$. We must show that $T(n+1) = c \cdot 2^{n+1}$

$$T(n+1) = 2T(n) + O(1) =$$
$$T(n+1) = 2(c \cdot 2^n) + d =$$
$$T(n+1) = c \cdot 2^{n+1} + d =$$
$$T(n+1) = O(2^n)$$

b

Substitution Proof:
$T(n) = 2T(n-1) + 1$ has the solution $T(n) = O(2^n)$ Assume that $T(n) \leq c \cdot 2^n - k$. We must show that $T(n+1) = c \cdot 2^{n+1}$

$$T(n+1) = 2T(n) + O(1) =$$
$$T(n+1) = 2(c \cdot 2^n - k) + d =$$
$$T(n+1) = c \cdot 2^{n+1} - 2k + d =$$
$$T(n+1) = O(2^n)$$

Question 2

a

The running time of randomized quicksort algorithm for A runs in $\Theta(n^2)$. This can be seen simply through observing that in each recursive call of RANDOMIZED-PARTITION, a swap occurs between the selected pivot and every element.
Thus we must solve subproblems of sizes $n + n - 1 + n - 2 + ... + 2 + 1 = \frac{n(n-1)}{2} = \Theta(n^2)$

b

A = $(4, 9, 5, 8, 2, 6)$ before the execution of the algorithm.

1 i= RANDOM(1, 6) # In this case we will assume i to be 3

2 exchange A[6] with A[3] # A now looks like $(4, 9, 6, 8, 2, 5)$

3 return PARTITION (A, p, r) # $(4, 2, 5, 6, 9, 8)$

Question 3

We must compute the expected number of inversions in A. Let X_{ij} be an *indicator random variable*. We define it as follows

$$X_{ij} = \begin{cases} 1 \text{ if } & A[i] > A[j] \\ 0 \text{ else} \end{cases}$$

$E[X_{i,j}] = \frac{1}{2}$
$E[x] = E\left[\sum_{i=1}^{n} \sum_{j=i+1}^{n} X_{i,j}\right]$ by linearity of expectation. Note that we only take the pairs with $i < j$

$$E[x] = \left(\frac{n(n-1)}{2}\right) E\left[X_{i,j}\right]$$
$$E[x] = \left(\frac{n(n-1)}{2}\right)\left(\frac{1}{2}\right)$$
$$E[x] = \frac{n(n-1)}{4}$$

.

Question 4

COW, DOG, SEA, RUG, ROW, MOB, BOX, TAB, BAR, EAR, TAR, DIG, BIG, TEA, NOW, FOX
Looking at the ending digit:
SEA, TEA, MOB, TAB, DOG, RUG, DIG, BIG, BAR, EAR, TAR, COW, ROW, NOW, BOX, FOX
Now the middle digit:
TAB, BAR, EAR, TAR, SEA, TEA, DIG, BIG, MOB, DOG, COW, ROW, NOW, BOX, FOX, RUG
Finally, the beginning digit, leading to a sorted array:
BAR, BIG, BOX, COW, DIG, DOG, EAR, FOX, MOB, NOW, ROW, RUG, SEA, TAB, TAR, TEA

Question 5

We must pre-process the data similar to bin sort. Each bucket $i \in [1...k]$will contain the interval $[\frac{a-b}{k} \cdot (i-1), \frac{a-b}{k} \cdot i]$. Place element of value v into bucket $\frac{v}{k}$. Note that this may lead to a double insertion in adjacent bins, but this will not be an issue. This takes $O(n+k)$ **Now for the algorithm:**
Return the size of bucket $\left[\frac{a}{k}, \frac{b}{k}\right]$