ADVANCED DATA STRUCTURES
HOMEWORK 1
SOLUTION

1) a- $T(n) = T(n-1) + 1/n$

Assume $T(1) = 1$, then unroll the recurrence:

$$T(n) = T(n-1) + \frac{1}{n}$$
$$= \frac{1}{n} + \frac{1}{n-1} + \frac{1}{n-2} + \cdots + \frac{1}{2} + \frac{1}{T(1)}$$

Lets bound the sum using integrals

$$T(n) = \sum_{k=1}^{n} \frac{1}{k} \geq \int_1^{n+1} \frac{1}{x}\,dx = \ln(n+1) \geq \ln n = \Omega(\ln n)$$

$$T(n) = \sum_{k=1}^{n} \frac{1}{k} = 1 + \sum_{k=2}^{n} \frac{1}{k} \leq 1 + \int_1^{n} \frac{1}{x}\,dx = 1 + \ln n = O(\ln n)$$

$$T(n) = \Theta(\ln n) = \Theta(\lg n)$$

b- $T(n) = 2T(n/2) + n\lg n$

Assume $n = 2^m$ for some m and $T(1) = c$, then unroll the recurrence:

$$T(n) = 2T\left(\frac{n}{2}\right) + n\lg n$$
$$= 2\left(2T\left(\frac{n}{4}\right) + \frac{n}{2}\lg\frac{n}{2}\right) + n\lg n$$
$$= 2\left(2\left(2T\left(\frac{n}{8}\right) + \frac{n}{4}\lg\left(\frac{n}{4}\right)\right) + \frac{n}{2}\lg\frac{n}{2}\right) + n\lg n$$
$$= 2\left(2\left(\cdots 2\left(2\left(2T(1) + 2\lg 2\right) + 4\lg 4\right) + \cdots + \frac{n}{4}\lg\frac{n}{4}\right) + \frac{n}{2}\lg\frac{n}{2}\right) + n\lg n$$
$$= n\left(c + \lg 2 + \lg 4 + \cdots + \lg\frac{n}{4} + \lg\frac{n}{2} + \lg n\right)$$
$$= n\left(c + \sum_{k=1}^{\lg n} k\right)$$

$$= n\left(c + \frac{\lg^2 n + \lg n}{2}\right)$$

$$= \frac{1}{2} n \lg^2 n + \frac{1}{2} n \lg n + cn$$

$$= \Theta(n \lg^2 n)$$

c- $5T(n/5) + n/\lg n$

Assume $n = 5^m$ for some m and $T(1) = c$, then unroll the recurrence:

$$= 5\left(5T\left(\frac{n}{25}\right) + \frac{n}{5}/\lg\frac{n}{5}\right) + n/\lg n$$

$$= 5\left(5\left(\cdots 5\left(5\left(5T(1) + 5/\lg 5\right) + 25/\lg 25\right) + \cdots + \frac{n}{25}/\lg\frac{n}{25}\right) + \frac{n}{5}/\lg\frac{n}{5}\right)$$
$$+ n/\lg n$$

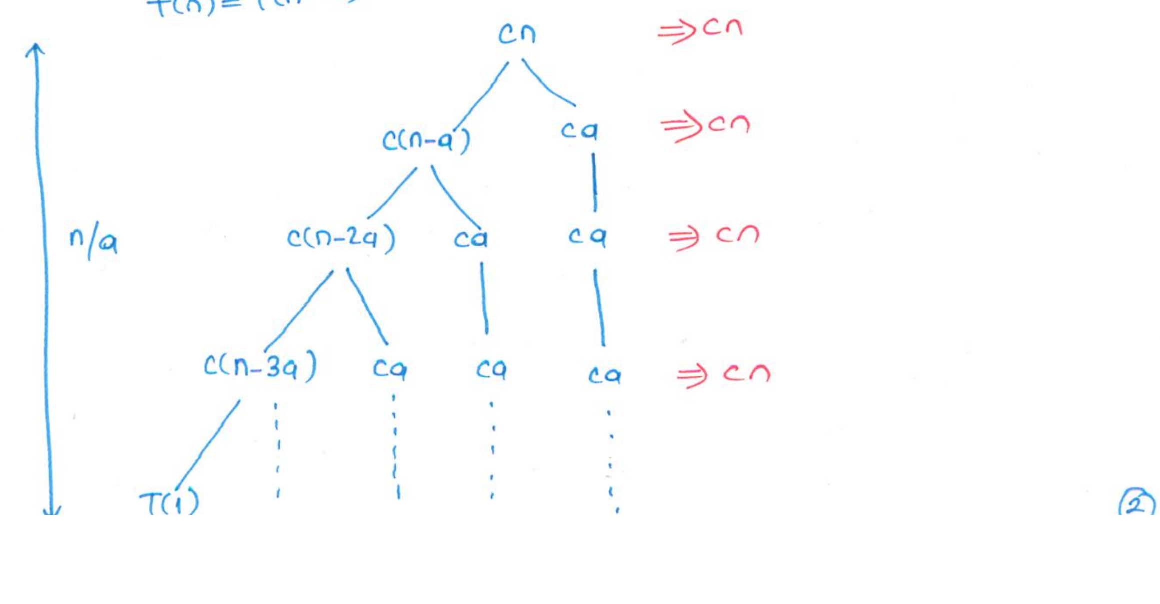
$$= n\left(c + 1/\lg 5 + 1/\lg 5^2 + \cdots + 1/\lg\frac{n}{5^2} + 1/\lg\frac{n}{5} + 1/\lg n\right)$$

$$= n\left(c + \sum_{k=1}^{\log_5 n} \frac{1}{\lg 5^k}\right)$$

$$= nc + \frac{n}{\lg 5} \sum_{k=1}^{\log_5 n} \frac{1}{k}$$

$$= \Theta(n \lg\lg n)$$

2) Use and draw a recursion tree to give an asymptotically tight solution to the recurrence $T(n) = T(n-a) + T(a) + cn$ where $a \geq 1$ and $c > 0$ are constants



The tree ends at $n=1$ which happens when $n-ka=1$

i.e., $k=(n-1)/a \approx n/a$ levels.

The total sum is $T(n)=(n/a)cn=O(n^2)$

Now verify that $T(n)=O(n^2)$ that $T(n)<dn^2$

Assume $T(k)<dk^2$ for all $k<n$, then show it for $k=n$:

$$\begin{aligned} T(n) &= T(n-a)+T(a)+cn \\ &< d(n-a)^2+da^2+cn \\ &= dn^2-2adn+2da^2+cn \\ &= dn^2-(2adn-2da^2-cn) \end{aligned}$$

In order $(2adn-2da^2-cn)$ to be greater than 0, $d=(c+1)/2a$

then we get $T(n)<dn^2-(n-ac-a)$ which will be less than dn^2 for large values of n, $T(n)=O(n^2)$.

3)
```
int list[1...n];
int even_sum=0, odd-sum=0, total-product=1;
alg(list, even-sum, odd-sum, total_product)
{   if (list→element is odd)
        odd-sum←odd-sum+list→element;
        list←list→next;
        alg(list, even-sum, odd-sum, total_product);
    if (list→element is even)
        even-sum←even-sum+list→element;
        list←list→next;
        alg(list, even-sum, odd-sum, total_product);
    if (list→element is null)
        total_product=even-sum*odd-sum;

    return total-product
}
```

Recursion can be represented as:

$$T(n) = T(n-1) + c$$

$$\left.\begin{aligned} T(n) &= c \\ T(n-1) &= c \\ &\vdots \\ T(1) &= c \end{aligned}\right\} n$$

$$n \cdot c \qquad T(n) = \Theta(n)$$

4) unsortedSearch (A, t, p, q)

1 if $q - p < 1$
2 if $A[p] = t$ return 1 else return 0
3 if unsortedSearch $(A, t, p, \lfloor \frac{p+q}{2} \rfloor) = 1$ return 1
4 if unsortedSearch $(A, t, \lfloor \frac{p+q}{2} \rfloor + 1, q) = 1$ return 1
return 0

The first two lines take constant time, call it c. The next two lines recursively call unsortedSearch on inputs of size $n/2$. Therefore, the worst-case asymptotic complexity is

$$T(n) = 2T(n/2) + c$$

Master theorem (case 1), we see that $T(n) = \Theta(n)$

5) Let A_{ij}, $i<j$ be an indicator random variable for the event where $X[i] > X[j]$. $\Rightarrow A_{ij} = I\{X[i] > X[j]\}$ for $1 \le i < j \le n$

$Pr\{A_{ij} = 1\} = 1/2$ (bcz the probability that the first is bigger than the second is $1/2$. $E[A_{ij}] = 1/2$

Let A be the random variable denoting the total number of inverted pairs in the array. A is the sum of all A_{ij} that meet the constrain $1 \le i < j \le n$

$$A = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} A_{ij}$$

Take the expectation of both sides

$$E[A] = E\left[\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} A_{ij}\right]$$

Then,

$$= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} E[A_{ij}]$$

$$= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} 1/2 = \binom{n}{2} \frac{1}{2} = \frac{n(n-1)}{2} \cdot \frac{1}{2} = \frac{n(n-1)}{4}$$

Therefore, the expected number of inverted pair is $n(n-1)/4$ or $O(n^2)$