Anar Rzayev   CS300: Homework 2   ID: 20190788

1) a) Let $C = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}$ with $\begin{cases} C_{11} = A_{11}B_{11} + A_{12}B_{21} \\ C_{12} = A_{11}B_{12} + A_{12}B_{22} \\ C_{21} = A_{21}B_{11} + A_{22}B_{21} \\ C_{22} = A_{21}B_{12} + A_{22}B_{22} \end{cases}$

$P_2 + P_3 = A_{11}B_{11} + A_{12}B_{21} =$
$= C_{11} \Rightarrow \boxed{C_{11} = P_2 + P_3}$

$T_1 + P_5 + P_6 = (P_1 + P_2) + P_5 + P_6 = (S_2S_4 + A_{11}B_{11}) + S_1S_3 +$
$+ (A_{12} - S_2)B_{22}$ and since S_2, S_4, S_1, S_3 - are $(n/2)\times(n/2)$ matrices (S_1 - sum of $(n/2)\times(n/2)$ matrices, and similarly for others)
$S_2S_4 = (S_1 - A_{11})(B_{22} - S_3) = S_1B_{22} - A_{11}B_{22} - S_1S_3 + A_{11}S_3$, with all multiplications are possible because all of them are $(n/2)\times(n/2)$ matrices $\Rightarrow T_1 + P_5 + P_6 = S_1B_{22} - A_{11}B_{22} - S_1S_3 +$
$+ A_{11}S_3 + A_{11}B_{11} + S_1S_3 + A_{12}B_{22} - S_2B_{22} =$
$= A_{11}B_{11} + A_{12}B_{22} - A_{11}B_{22} + S_1B_{22} - B_{22}(S_1 - A_{11}) +$
$+ A_{11}(B_{12} - B_{11}) = A_{11}B_{11} + A_{12}B_{22} - A_{11}B_{22} + S_1B_{22} - S_1B_{22} +$
$+ A_{11}B_{22} + A_{11}B_{12} - A_{11}B_{11} = A_{12}B_{22} + A_{11}B_{12}$, where all summations, differences, and products are possible because all sizes of matrices are $(n/2)\times(n/2) \Rightarrow \boxed{T_1 + P_5 + P_6 = A_{12}B_{22} + A_{11}B_{12}}$
$\Rightarrow \boxed{C_{12} = T_1 + P_5 + P_6}$

$T_2 - P_7 = T_1 + P_4 - P_7 = P_1 + P_2 + P_4 - P_7 = S_2S_4 + A_{11}B_{11} + P_4 - P_7$

$T_2 - P_7 = \delta_2\delta_4 + A_{11}B_{11} + (A_{11}-A_{21})(B_{22}-B_{12}) + -A_{22}(\delta_4 - B_{21})$
$= (\delta_1 - A_{11})(B_{22}-\delta_3) + A_{11}B_{11} + A_{11}B_{22} - A_{21}B_{22} - A_{11}B_{12} +$
$+ A_{21}B_{12} - A_{22}(B_{22} - \delta_3 - B_{21}) = (A_{21}+A_{22}-A_{11})(B_{22}-B_{12}+B_{11})$
$+ A_{11}B_{11} + A_{11}B_{22} - A_{21}B_{22} - A_{11}B_{12} + A_{21}B_{12} - A_{22}B_{22} +$
$+ A_{22}(B_{12}-B_{11}) + A_{22}B_{21} = \cancel{A_{21}B_{22}} + \cancel{A_{22}B_{22}} - \cancel{A_{11}B_{22}} -$
$- \cancel{A_{21}B_{12}} - \cancel{A_{22}B_{12}} + \cancel{A_{11}B_{12}} + A_{21}B_{11} + \cancel{A_{22}B_{11}} - \cancel{A_{11}B_{11}} +$
$+ \cancel{A_{11}B_{11}} + \cancel{A_{11}B_{22}} - \cancel{A_{21}B_{22}} - \cancel{A_{11}B_{12}} + \cancel{A_{21}B_{12}} - \cancel{A_{22}B_{22}} +$
$+ \cancel{A_{22}B_{12}} - \cancel{A_{22}B_{11}} + A_{22}B_{21} = A_{21}B_{11} + A_{22}B_{21}$, where we again note that all multiplications, sums, differences are allowed because of $(n/2)\times(n/2)$ dimensions $\Rightarrow$ $T_2 - P_7 = A_{21}B_{11} + A_{22}B_{21}$
$\Rightarrow \boxed{T_2 - P_7 = C_{21}}$ (*)

$T_2 + P_5 = T_1 + P_4 + P_5 = P_1 + P_2 + P_4 + P_5 = \delta_2\delta_4 + A_{11}B_{11} +$
$+ (A_{11}-A_{21})(B_{22}-B_{12}) + \delta_1\delta_3 = (\delta_1 - A_{11})(B_{22}-\delta_3) +$
$+ A_{11}B_{11} + A_{11}B_{22} - A_{21}B_{22} - A_{11}B_{12} + A_{21}B_{12} + \delta_1\delta_3 =$
$= \delta_1 B_{22} - \cancel{A_{11}B_{22}} - \cancel{\delta_1\delta_3} + A_{11}\delta_3 + A_{11}B_{11} + \cancel{A_{11}B_{22}} - A_{21}B_{22} -$
$- A_{11}B_{12} + A_{21}B_{12} + \cancel{\delta_1\delta_3} = (A_{21}+A_{22})B_{22} + A_{11}(B_{12}-B_{11}) +$
$+ A_{11}B_{11} - A_{21}B_{22} - A_{11}B_{12} + A_{21}B_{12} = \cancel{A_{21}B_{22}} + A_{22}B_{22} +$
$+ \cancel{A_{11}B_{12}} - \cancel{A_{11}B_{11}} + \cancel{A_{11}B_{11}} - \cancel{A_{21}B_{22}} - \cancel{A_{11}B_{12}} + A_{21}B_{12} =$
$= A_{21}B_{12} + A_{22}B_{22} \Rightarrow \boxed{T_2 + P_5 = A_{21}B_{12} + A_{22}B_{22}}$ Hence,

$\boxed{C_{22} = T_2 + P_5}$ with considering all operations were legal

$$\Rightarrow \boxed{C = \begin{pmatrix} P_2 + P_3 & T_1 + P_5 + P_6 \\ T_2 - P_7 & T_2 + P_5 \end{pmatrix}} \quad \boxed{v}$$

b) $S_1 = 7 + 5 = 12$

$S_2 = 12 - 4 = 8$

$S_3 = 8 - 6 = 2$

$S_4 = 3 - 2 = 1$

$P_1 = 8 \cdot 1 = 8$

$P_2 = 4 \cdot 6 = 24$

$P_3 = 2 \cdot 1 = 2$

$P_4 = (4-7)(3-8) = (-3)(-5) = 15$

$P_5 = 12 \cdot 2 = 24$

$P_6 = (2-8)3 = -18$

$P_7 = 5(1-1) = 0$

$T_1 = 8 + 24 = 32$

$T_2 = 32 + 15 = 47$

$$\begin{pmatrix} 4 & 2 \\ 7 & 5 \end{pmatrix} \begin{pmatrix} 6 & 8 \\ 1 & 3 \end{pmatrix} = \begin{pmatrix} 24 + 2 & 32 + 6 \\ 42 + 5 & 56 + 15 \end{pmatrix} = \begin{pmatrix} 26 & 38 \\ 47 & 71 \end{pmatrix} \text{ and}$$

$P_2 + P_3 = 24 + 2 = 26$ ✓ $\quad T_1 + P_5 + P_6 = 32 + 24 - 18 = 32 + 6 = 38$ ✓

$T_2 - P_7 = 47 - 0 = 47$ ✓ $\quad T_2 + P_5 = 47 + 24 = 71$ ✓

Hence, we used above's method to find and confirm product ∎

c) First, we divide the input matrices A and B and output matrix C into $(n/2) \times (n/2)$ submatrices. This step takes $\Theta(1)$ time by index calculation

Then, we create 4 matrices S_1, S_2, S_3, S_4, each of which is $(n/2) \times (n/2)$ and is the sum or difference of two matrices (i.e. S_1 - sum of A_{21}, A_{22}; S_2 - difference of S_1 and A_{11}; S_3 - difference of B_{12}, B_{11}; S_4 - difference of B_{22} and S_3)

created in initial and this step. We can create all 4 matrices in $\Theta(n^2)$ time.
Using the submatrices created in very initial step and 4 matrices created in previous step, we recursively compute 7 matrix products $P_1, P_2, \dots, P_7$, where each P_i is $(n/2)\times(n/2)$ and $A_{11}-A_{21}$, $B_{22}-B_{12}$, $A_{12}-S_2$, S_4-B_{21}, and others are all $(n/2)\times(n/2)$ dimensions. Therefore, computing $A_{11}-A_{21}$, $B_{22}-B_{12}$, $A_{12}-S_2$, S_4-B_{21} requires $\Theta(n^2)$ time and others have been already computed (such as $S_2, S_4, A_{11}, B_{11}, \dots$)
Therefore, we are left with 14 $(n/2)\times(n/2)$ matrices which should be multiplied (pair-by-pair), requiring us 7 number of matrix multiplications
After that, we find T_1 and T_2, by summing up $(n/2)\times(n/2)$ matrices P_1, P_2 and T_1, P_4. This requires $\Theta(n^2)$ time since we just add up matrices.
Finally, we conclude the desired submatrices $C_{11}, C_{12}, C_{21}, C_{22}$ of the result matrix C by adding and subtracting various combinations of the P_i and T_j matrices, where all of them are in $(n/2)\times(n/2)$ dimensions. We can compute all 4 submatrices in $\Theta(n^2)$ time
Coming to strassen's method, it's not more efficient, and mostly it's the same. Since we have 7 matrix multiplications of size $(n/2)\times(n/2)$ in both methods, we'll have $7T\left(\frac{n}{2}\right)$ in both (meaning 7 recursive calls will prevail in both approaches)

We can observe there are 4 summations(differences) to compute S_i, 4 summations (differences) during the computation of P_j (i.e. $A_{11}-A_{21}$, $B_{22}-B_{12}$, $A_{12}-S_2$, S_4-B_{21}), and 2 sums to find T_1 and T_2 (P_1 & P_2 and T_1 & P_4). At the end, there'll be 5 summations(differences) to conclude the submatrices for matrix C => overall, there are 15 sums or differences in this approach. In Strassen's approach, we evaluate 10 sums(differences) for S_i, and 8 sums(differences) among P_i's to evaluate submatrices $C_{11}, C_{12}, C_{21}, C_{22}$. => overall, there are 18 sums or differences in Strassen's method

So, we can see # of matrix multiplications are same, but # of summations (differences) in new method is less than that of Strassen's method. When we discuss about running time, it's essentially same as all sums (diffs) will be subsumed as $\Theta(n^2)$ but we should admit that # of sums (diff.s) in new method is less than that of Strassen's method (15 vs 18) => A little bit (for small inputs) efficient than Strassen's method as # of sums(diff.s) is less, but matrix multiplications are same ✓ (For large inputs it's mostly same)

$\boxed{\text{Efficient in terms of summations (diff.s) but same in terms of matrix multiplications (For large inputs, it's same)}}$

d) As discussed in previous section, all $\Theta(n^2)$'s will be subsumed when we operate summations(diff.s). The desired is to know # of matrix multiplications => 7

$T(n) = 7T\left(\frac{n}{2}\right) + \Theta(n^2)$ for $n > 1$ (we assume $T(1) = \Theta(1)$)

$$T(n) = \begin{cases} \Theta(1), & \text{if } n=1 \\ 7T\left(\frac{n}{2}\right) + \Theta(n^2), & \text{if } n>1 \end{cases}$$

e) For $n>1 \Rightarrow T(n) = 7T\left(\frac{n}{2}\right) + \Theta(n^2)$ and using Master's method
$a=7$, $b=2$, $f(n) = \Theta(n^2) \Rightarrow n^{\log_b a} = n^{\lg 7}$ with $2.8 < \lg 7 < 2.81 \Rightarrow$
$f(n) = O(n^{\lg 7 - \varepsilon})$ for $\varepsilon = 0.8 \Rightarrow$ case 1 applies, and we have

$$\boxed{T(n) = \Theta(n^{\lg 7})}$$

f) For each $n \in \mathbb{N}$, we take m such that $2^{m-1} \leq n < 2^m$, where it's possible always to find such m (In fact, every integer n lies on the interval $[2^{m-1}, 2^m)$ for some integer m) Considering n is not an exact power of 2, we claim it's feasible to use the above method by extending our matrix to an $2^m \times 2^m$ matrix and pad it with zeros where needed (basically, filling in the empty entries with zeros) So, we pad out the input matrices to be powers of two and run our algorithm. Padding out to the next largest power of two (i.e. in our case, 2^m) will at most double the value of n, because each power of 2 is off from each other by a factor of two $\Rightarrow (2^m)^{\lg 7} \leq (2n)^{\lg 7} = 7n^{\lg 7} \in O(n^{\lg 7})$ and $(2^m)^{\lg 7} \geq n^{\lg 7} \in \Omega(n^{\lg 7})$. Hence, we've that running time $\longrightarrow \boxed{\Theta(n^{\lg 7})}$ ■

Scanned with CamScanner

2) a) When we were discussing average-case behavior of quicksort, there was made an assumption that all permutations of the input are equally likely. However, it's not always the case. So, by adding randomization to an algorithm, we get good expected performance over all inputs.

We'll perform random sampling, a randomization technique, for quicksort. Instead of using $A[r]$ or $A[p]$ as the pivot, we'll select a randomly chosen element from subarray $A[p..r]$. Initially we swap $A[r]$ (or $A[p]$) with an element chosen at random from $A[p..r]$. By randomly sampling the range $p, \dots, r$, we're sure that pivot element $x = A[r]$ (or $A[p]$) is equally likely to be any of the $(r-p+1)$ elements in the subarray. As we randomly choose pivot, it's reasonable to expect the split of the input array mostly well-balanced on average.

New Partition function → Randomized-Partition: Randomly choose element, swap it with last (first) element, and then proceed normal (well-known) partitioning as usual.

New QuickSort function → Randomized-QuickSort: New quicksort calls Randomized-Partition instead of Partition

Note: It should be also mentioned that we analyze the expected running time of a randomized algorithm (not worst-case) since it represents the more typical time cost. Also, we're doing the expected run time over the possible randomness used during computation

b) Since pivot is selected as a random element in the array, (size $= n$), probabilities of any particular element being selected are all equal $\Rightarrow$ are all $= \frac{1}{n}$ (since summations of probabilities are $= 1$)

Using definition of $E[X_i] \Rightarrow E[X_i] = 0 \cdot \Pr\{i^{th}$ smallest is not chosen as a pivot$\}$ $+ 1 \cdot \Pr\{i^{th}$ smallest is chosen as a pivot$\} = \Pr\{i^{th}$ smallest element is chosen as a pivot$\}$ $= \frac{1}{n}$. from previous result $\Rightarrow$ $\boxed{E[X_i] = \frac{1}{n}}$

c) The key idea is to use linearity of expectation over all of the events X_i. Assume for a particular case, X_i is true, then in one sub-array $\rightarrow$ length $= i-1$ and on the other sub-array, length $= n-i \Rightarrow$ Moreover, we'll also need linear time to run the partition procedure (as discussed in "for" loop part of partition)

$$T(n) = \begin{cases} T(0) + T(n-1) + \Theta(n), & \text{if } 1^{th} \text{ smallest element is chosen as a pivot} \\ T(1) + T(n-2) + \Theta(n), & \text{if } 2^{nd} \text{ smallest is chosen as a pivot} \\ \vdots & \\ T(n-1) + T(0) + \Theta(n), & \text{if } n^{th} \text{ smallest is pivot} \end{cases}$$

$$\Rightarrow E[T(n)] = E\left[\sum_{q=1}^{n} X_q \left(T(q-1) + T(n-q) + \Theta(n)\right)\right]$$

d) Using linearity of expectation and Independence of X_q from other random choices $\Rightarrow$ $E[T(n)] = \sum_{q=1}^{n} E\left[X_q \left(T(q-1) + T(n-q) + \Theta(n)\right)\right]$

$$= \sum_{q=1}^{n} E[X_q] \cdot E\left[T(q-1) + T(n-q) + \Theta(n)\right]$$

using the fact that $E[X_i] = \frac{1}{n}$ and linearity of expectation

$$E[T(n)] = \frac{1}{n}\sum_{q=1}^{n} E[T(q-1)] + \frac{1}{n}\sum_{q=1}^{n} E[T(n-q)] + \frac{1}{n}\underbrace{\sum_{q=1}^{n}\Theta(n)}_{n\Theta(n)} =$$

$$= \frac{1}{n}\sum_{q=1}^{n} E[T(q-1)] + \frac{1}{n}\sum_{q=1}^{n} E[T(n-q)] + \Theta(n)$$ where we observe

$$\sum_{q=1}^{n} E[T(n-q)] = E[T(n-1)] + \dots + E[T(0)] = \sum_{q=1}^{n} E[T(q-1)] =$$

$$= E[T(0)] + \dots + E[T(n-1)] \Rightarrow E[T(n)] = \frac{2}{n}\sum_{q=1}^{n} E[T(q-1)] + \Theta(n)$$

$$= \frac{2}{n}\sum_{q=0}^{n-1} E[T(q)] + \Theta(n) = \frac{2}{n}E[T(0)] + \frac{2}{n}\sum_{q=1}^{n-1} E[T(q)] + \Theta(n)$$

$$= \frac{2}{n}E[T(0)] + \frac{2}{n}E[T(1)] + \frac{2}{n}\sum_{q=2}^{n-1} E[T(q)] + \Theta(n)$$ and

since $\frac{2}{n}\left(E[T(0)] + E[T(1)]\right)$ can be subsumed in $\Theta(n) \Rightarrow$

$$\boxed{E[T(n)] = \frac{2}{n}\sum_{q=2}^{n-1} E[T(q)] + \Theta(n)}$$ ✓

e) $\sum_{k=2}^{n-1} k\lg k = \sum_{k=2}^{\lceil \frac{n}{2}\rceil - 1} k\lg k + \sum_{k=\lceil \frac{n}{2}\rceil}^{n-1} k\lg k$ with $\lg\left(\lceil \frac{n}{2}\rceil - 1\right) \le \lg n - 1$

(If $n = 2k \Rightarrow \lg(k-1) < \lg k = \lg\left(\frac{n}{2}\right) = \lg n - 1$, $n = 2k+1 \Rightarrow \lceil \frac{n}{2}\rceil = \lceil k + \frac{1}{2}\rceil = k+1$.

$\lg\left(\lceil \frac{n}{2}\rceil - 1\right) = \lg(k) < \lg\left(k + \frac{1}{2}\right) = \lg\left(\frac{n}{2}\right) = \lg n - 1$ ✓ $) \Rightarrow \sum_{k=2}^{\lceil \frac{n}{2}\rceil - 1} k\lg k \le$

$\le \sum_{k=2}^{\lceil \frac{n}{2}\rceil - 1} k\lg\left(\lceil \frac{n}{2}\rceil - 1\right) \le (\lg n - 1)\sum_{k=2}^{\lceil \frac{n}{2}\rceil - 1} k$ and $\sum_{k=\lceil \frac{n}{2}\rceil}^{n-1} k\lg k \le \sum_{k=\lceil \frac{n}{2}\rceil}^{n-1} k\cdot\lg(n-1)$

$< \sum_{k=\lceil \frac{n}{2}\rceil}^{n-1} k\cdot\lg n \Rightarrow \sum_{k=2}^{n-1} k\lg k < (\lg n - 1)\sum_{k=2}^{\lceil \frac{n}{2}\rceil - 1} k + \lg n\cdot\sum_{k=\lceil \frac{n}{2}\rceil}^{n-1} k$ with having

$$\sum_{k=2}^{n-1} k\lg k < \lg n\left(\sum_{k=2}^{\lceil \frac{n}{2}\rceil-1} k + \sum_{k=\lceil \frac{n}{2}\rceil}^{n-1} k\right) - \sum_{k=2}^{\lceil \frac{n}{2}\rceil-1} k = \lg n \cdot \sum_{k=2}^{n-1} k - \sum_{k=2}^{\lceil \frac{n}{2}\rceil-1} k =$$

$$= \lg n \cdot \left(\frac{(n-1)n}{2} - 1\right) - \left(\frac{(\lceil \frac{n}{2}\rceil-1)\lceil \frac{n}{2}\rceil}{2} - 1\right)$$

$$\cancel{- \frac{(\lceil \frac{n}{2}\rceil-1)\lceil \frac{n}{2}\rceil}{2} + 1 < \frac{n^2\lg n}{2} - \left(\frac{(\lceil \frac{n}{2}\rceil-1)\lceil \frac{n}{2}\rceil}{2} - 1\right) \overset{?}{\le}}$$

$\cancel{\overset{?}{\le} \frac{n^2\lg n}{2} - \frac{1}{8}n^2}$, ~~hence we should prove~~ $\cancel{\frac{(\lceil \frac{n}{2}\rceil-1)\lceil \frac{n}{2}\rceil}{2} - 1 \overset{?}{\ge} \frac{1}{8}n^2}$

$\cancel{n=2k \Rightarrow \frac{(k-1)k}{2} - 1 \overset{?}{\ge} \frac{1}{8}\cdot 4k^2}$ ~~or~~ $\cancel{\frac{k^2-k-2}{2} \overset{?}{\ge}}$

$$\sum_{k=2}^{n-1} k\lg k < \lg n \cdot \left(\frac{n^2-n}{2} - 1\right) - \left(\frac{(\lceil \frac{n}{2}\rceil-1)\lceil \frac{n}{2}\rceil}{2} - 1\right) =$$

$$= \cancel{\frac{n^2\lg n}{2}} - \frac{n\lg n}{2} - \lg n - \left(\frac{(\lceil \frac{n}{2}\rceil-1)\lceil \frac{n}{2}\rceil}{2} - 1\right) \overset{?}{\le} \cancel{\frac{n^2\lg n}{2}} - \frac{n^2}{8}$$

so, we should prove $\Rightarrow \frac{n\lg n}{2} + \lg n + \frac{(\lceil \frac{n}{2}\rceil-1)\lceil \frac{n}{2}\rceil}{2} - 1 \overset{?}{\ge} \frac{n^2}{8}$

$\underline{n=2k} \Rightarrow k\lg(2k) + \lg(2k) + \frac{(k-1)k}{2} - 1 = k(1+\lg k) + \cancel{1} + \lg k +$

$+ \frac{(k-1)k}{2} - \cancel{1} = k + k\lg k + \lg k + \cancel{\frac{k^2}{2}} - \frac{k}{2} \overset{?}{\ge} \frac{4k^2}{8} = \cancel{\frac{k^2}{2}} \Rightarrow$

$\lg k(k+1) + \frac{k}{2} \overset{?}{\ge} 0$ which's obviously true ✓. $\underline{n=2k+1} \Rightarrow \lceil \frac{n}{2}\rceil = k+1$

$$\frac{(2k+1)\lg(2k+1)}{2} + \lg(2k+1) + \frac{k(k+1)}{2} - 1 \overset{?}{\ge} \frac{(2k+1)^2}{8} = \frac{4k^2+4k+1}{8}$$

$$LHS > \frac{(2k+1)\lg(2k)}{2} + \lg(2k) + \cancel{\frac{k^2}{2}} + \cancel{\frac{k}{2}} - 1 \overset{?}{\ge} \cancel{\frac{k^2}{2}} + \cancel{\frac{k}{2}} + \frac{1}{8}$$

$LHS \geq \frac{(2k+1)(1+\lg k)}{2} + \cancel{1} + \lg k + \cancel{\frac{k^2}{2}} + \cancel{\frac{k}{2}} - \cancel{1} \geq \cancel{\frac{k^2}{2}} + \cancel{\frac{k}{2}} + \frac{1}{8}$

$\frac{2k+1}{2} + \frac{(2k+1)\lg k}{2} + \lg k \geq \frac{1}{8}$, since $n = 2k+1 \geq 3 \Rightarrow \underline{k \geq 1}$ and

$\frac{2k+1}{2} \geq \frac{3}{2} > \frac{1}{8} \Rightarrow \checkmark$ Hence for $\underline{n = 2k+1 \Rightarrow \checkmark}$ meaning that

$$\sum_{k=2}^{n-1} k \lg k < \lg n \cdot \left(\frac{n^2 - n - 1}{2}\right) - \left(\frac{(\lceil \frac{n}{2} \rceil - 1)\lceil \frac{n}{2} \rceil - 1}{2}\right) \leq \frac{n^2 \lg n}{2} - \frac{n^2}{8}$$

meaning that $\boxed{\sum_{k=2}^{n-1} k \lg k \leq \frac{n^2 \lg n}{2} - \frac{n^2}{8}}$ ■

$\boxed{\text{for } n \geq 3}$

f) $E[T(n)] = \frac{2}{n} \sum_{q=2}^{n-1} E[T(q)] + \Theta(n)$, we'll prove that

$E[T(n)] \leq a n \lg n$ for constant $a > 0$.

We choose a large enough so that $a n \lg n$ dominates $E[T(n)]$ for sufficiently small $n \geq 2$. Using Substitution Method and putting inductive hypothesis $\Rightarrow E[T(n)] \leq \frac{2}{n} \sum_{q=2}^{n-1} a \cdot q \lg q +$

$+ \Theta(n) = \frac{2a}{n} \cdot \sum_{q=2}^{n-1} q \lg q + \Theta(n) \leq \frac{2a}{n} \cdot \left(\frac{n^2 \lg n}{2} - \frac{n^2}{8}\right) + \Theta(n) =$

$= a n \lg n - \frac{na}{4} + \Theta(n)$, by using inequality proved in e).

$E[T(n)] \leq a \cdot n \lg n - \left(\frac{an}{4} - \Theta(n)\right)$, if a is chosen large enough so that $\frac{an}{4}$ dominates the $\Theta(n) \Rightarrow E[T(n)] \leq a \cdot n \lg n$ and induction is proved

~~By proving $E[T(n)] \le$~~

By proving $E[T(n)] \le an\lg n$ for all sufficiently large n and for some constant a $(a>0)$ $\Rightarrow E[T(n)] = O(n\lg n)$. $\boxed{V}$ • E

~~By dominating, we mean that $\Theta(n) \le$ ... for some $c, > 0$~~

~~Q~~ Since $\Theta(n)$ is a tight bound, we can always find very large a such that $\frac{an}{4}$ will dominate $\Theta(n)$ term.