CS 38 Algorithms **April 19, 2023**

Problem Set 2

Codename: Jambul **Late Submission:** No

1 Problem 1

Consider the recurrence

$$T(n) = n\lg(n) + 2T(\lfloor n/2 \rfloor), T(1) = 1 \tag{1}$$

We will first show that the generalized master theorem does not provide us with the solution to this recurrence by considering the cases in which the theorem can provide us with a solution. Note that the general recurrence for the theorem is given by

$$T(n) = f(n) + \sum_{i=1}^{k} T(n/b_i).$$

Thus (1) can be written in this form if we set

$$f(n) = n\lg(n), k = 2, b_1 = b_2 = 2 \implies T(n) = n\lg(n) + T(n/2) + T(n/2).$$

We will now show that we cannot use the cases from the theorem to solve this recurrence. In case (a), we require that there exist $c > 1, n_0$ such that, for all $n > n_0, \sum_i f(n/b_i) \geq cf(n)$. In our case, we have $f(n) = n\lg(n)$, so we wish to find $c > 1, n_0$ so that

$$\sum_{i=1}^{2} \frac{n}{2}\lg(n/2) = n\lg(n/2) = n\lg(n) - n \geq c \cdot n\lg(n).$$

Fixing $c > 1$, this inequality requires that $0 < n < 2^{-1/(c-1)}$. Thus it is not the case that we can find some n_0 such that all $n > n_0$ satisfy the inequality above, as the inequality only holds for n below a finite bound.

In case (b), we require that $f(n) = n\lg(n) \in \Theta(n^w)$, for w the minimum root of $\sum_i b_i^{-w} = 2^{-w} + 2^{-w} = 1 \implies w = 1$. Trivially, $f(n) \notin \Theta(n)$, so this case is not applicable either.

Finally, in case (c), we seek $c < 1, n_0$ such that for all $n > n_0, \sum_i f(n/b_i) = n\lg(n) - n \leq cf(n) = cn\lg(n)$. Fixing $c < 1$, this inequality is satisfied only for $0 < n < 2^{-1/(c-1)}$, which again bounds the interval over which the inequality holds, and thus the theorem is not applicable to our recurrence.

We will now solve the recurrence. Take $n = 2^m$ for some $m \geq 0$. Then we have that

$$T(n = 2^m) = 2^m m\lg(2) + 2T(2^{m-1}) = m2^m + 2T(2^{m-1}).$$

Now, observe that

$$\begin{aligned} m2^m + 2T(2^{m-1}) &= m2^m + 2((m-1)2^{m-1} + 2T(2^{m-2})) \\ &= m2^m + (m-1)2^m + 4T(2^{m-2}), \end{aligned}$$

which is then

$$\begin{aligned} &m2^m + (m-1)2^m + 4((m-2)2^{m-2} + 2T(2^{m-3})) \\ &= m2^m + (m-1)2^m + (m-2)2^m + 8T(2^{m-3}), \end{aligned}$$

and when fully expanded,

$$T(2^m) = 2^m \sum_{i=1}^{m} i + 2^m T(2^0) = 2^m \left(1 + \sum_{i=1}^{m} i\right) = 2^m \left(1 + \frac{m(m+1)}{2}\right).$$

Thus, since we fixed $n = 2^m$, this is

$$T(n) = n\left(1 + \frac{\lg(n)(\lg(n)+1)}{2}\right) \in \boxed{\Theta(n \lg^2(n))}.$$

In particular, observe that for $c = \frac{1}{2}, C = 1, n^* = 4$, all $n \geq n^*$ have that $cf(n) \leq T(n) \leq Cf(n)$, for $f(n) = n\lg^2(n)$. Indeed, if $c = \frac{1}{2}$,

$$n + \frac{n\lg(n)}{2} + \frac{n\lg^2(n)}{2} \geq cn\lg^2(n) \iff n + \frac{n\lg(n)}{2} \geq 0,$$

which is of course true for all $n \geq 0$, and thus all $n \geq n^*$. Now, note that for $C = 1$,

$$n + \frac{n\lg(n)}{2} + \frac{n\lg^2(n)}{2} \leq Cn\lg^2(n) = n\lg^2(n) \iff n + \frac{n\lg(n)}{2} \leq \frac{n}{2}\lg^2(n).$$

At $n = n^* = 4$, this is

$$4 + 2\lg(4) = 8 \leq 2\lg^2(4) = 8.$$

Then, for all $n \geq n^*$, we will have that

$$(\lg(n))' = \frac{1}{n\ln(2)}, \left(\frac{\lg^2(n)}{2}\right)' = \frac{\lg(n)}{n\ln(2)} \geq \frac{1}{n\ln(2)}$$

for all $n \geq n^*$. Thus $\frac{n}{2}\lg^2(n)$ will grow faster than $n + \frac{n\lg(n)}{2}$ for $n \geq n^*$, and so the original inequality will hold for all $n \geq n^*$, with $C = 1$.

2 Problem 2

2.1 Part a

Proof. We start by defining some variables.

Let $L = [x_1, ..., x_n]$ be a list of length n, and $[x_i, x_j, x_k]$ some subsequence of L. Suppose, WLOG (we can always relabel the entries), that $x_i < x_j < x_k$, so our chosen pivot is $p = x_j$. Then denote $R = |\{x_l : x_l \in L\backslash p, x_l \leq p\}|$ the rank of p. Then we wish to show that, for $t \in [0,1]$,

$$P\left(\frac{R}{n} < t\right) = \int_0^t 6y(1-y)dy = 3t^2 - 2t^3.$$

We will start by computing $P(R = r | x_i < x_j < x_k)$. From Bayes' Law, we have

$$P(R = r | x_i < x_j < x_k) = P(x_i < x_j < x_k | R = r)\frac{P(R = r)}{P(x_i < x_j < x_k)}.$$

Trivially, we can set $P(R = r) = \frac{1}{n}$, since any single element in L has equal chance of being mapped to any rank $\{1, ..., n\}$, and $P(x_i < x_j = p < x_k) = \frac{1}{6}$, since there are $3! = 6$ permutations of $[x_i, x_j, x_k]$. Finally, note that, if we know the rank of the pivot is r, then $\frac{r}{n}$ is the percentage of entries in L which are less than or equal to $x_j = p$. Thus if we asked the chance that one randomly selected entry is less than p *and* one other randomly selected entry is greater than p, we would have that this is simply

$$\frac{r}{n}\left(1-\frac{r}{n}\right) = P(x_i < p < x_k | R = r).$$

Thus we may compute

$$P(R = r | x_i < x_j < x_k) = \frac{6}{n}\frac{r}{n}\left(1-\frac{r}{n}\right)$$

$$\implies P(R < r | x_i < x_j < x_k) = \sum_{m=0}^{r-1}\left(\frac{6m}{n^2} - \frac{6m^2}{n^3}\right)$$

$$= \frac{6}{n}\sum_{m=0}^{r-1} m + \frac{6}{n^3}\sum_{m=0}^{r-1} m^2 = \frac{6}{n}\frac{(r-1)(r)}{2} + \frac{6}{n^3}\frac{r-1}{6}(r+1)(2r+1).$$

Now setting $r = tn, t \in [0,1]$, this expression becomes

$$P(R < tn | x_i < x_j < x_k) = 3t^2 - \frac{3t}{n} - \frac{2t^3n^2 - t^2n - 2t^2n + t}{n^2} \xrightarrow[n\to\infty]{} 3t^2 - 2t^3,$$

as desired.

□

2.2 Part b

Proof. Carrying our notation from the last problem, let $X = \frac{R}{n}$. Note that

$$\int_0^1 6x(1-x)T(xn)dx = E(T(Xn)) = E(T(R)),$$

since we confirmed in part (a) that the density function of X is given by $f_X(x) = 6x(1-x)$. Thus we wish to show

$$T(n) \leq dn + 2E(T(Xn))$$

for some d.

We will first compute the worst-case number of comparisons in the original array L prior to splitting it into smaller subarrays. To find the first pivot, we will need to complete (at most) three comparisons (consider the case in which $x_i < x_j$ but $x_j > x_k$). Then, once we have found the pivot p, we will need to complete $n-3$ comparisons between p and the remaining elements in the list. Thus in total, we will be completing at most n comparisons to split the array for the very first time.

Now, once we have split the array, we will have two arrays, one of length xn, and one of length $(1-x)n$. Together, these subarrays are expected to take $E(T(Xn)) + E(T((1-X)n)) = 2E(T(Xn))$ comparisons to sort. Thus we obtain the bound

$$T(n) \leq dn + E(T(Xn))$$

by setting $d = 1$.

□

2.3 Part c

Suppose the solution to the recursion takes the form $T(n) = nb(c + \ln(n))$. (Where we have devised this brilliant solution is beside the point.)

Then, from (b), it would have to be the case that

$$nb(c + \ln(n)) \leq n + 2\int_0^1 6x(1-x)xnb(c + \ln(xn))dx$$

$$
\begin{aligned}
&= n + 2\left(\int_0^1 (6x^2nb - 6x^3nb)(c + \ln(xn))dx\right) \\
&= n + 12nb\left(\int_0^1 (x^2c + x^2\ln(xn) - x^3c - x^3\ln(xn))dx\right) \\
&= n + \frac{nb}{12}(12c + 12\ln(n) - 7) \\
\iff&\ nbc + nb\ln(n) \le n + nbc + nb\ln(n) - \frac{7}{12}nb \\
\iff&\ \frac{7}{12}b \le 1 \iff b \le \frac{12}{7},
\end{aligned}
$$

so indeed, $b = \frac{12}{7}$ satisfies the recurrence if we assume that the solution takes the form $T(n) = nb(c + \ln(n))$.

3 Problem 3

Algorithm 1. (Path of Least Disruption.) Denote our disruption matrix by d so $d(i,j)$ is the disruption measure of the pixel at $A(i,j)$. Then the solution to the problem can be written as follows.

1. Create an $m \times n$ table T with all entries initialized to $T(i,j) = \infty$.
2. Fill in the bottom row of the table (entries of the form $T(m,j)$) with the corresponding disruption measure $d(m,j)$.
3. For $i = 2$ to 1 by -1:
   (a) For $j = 1$ to m:
      i. If $j = 1$: Set $T(i,j) = \min\{T(i+1,j), T(i+1,j+1)\} + d(i,j)$
      ii. Else if $j = n$: Set $T(i,j) = \min\{T(i+1,j), T(i+1,j-1)\} + d(i,j)$
      iii. Else: Set $T(i,j) = \min\{T(i+1,j+1), T(i+1,j), T(i+1,j-1)\}$
4. Return $\min_j\{T(1,j)\}$

Proof of correctness. We will prove the correctness of the algorithm given above by backward induction on the indices i and j. In particular, we will show that for all $1 \le i \le m, 1 \le j \le n, T(i,j)$ corresponds to the minimum total disruption a seam starting at pixel $A(i,j)$ could have.

For our base case, consider entries of the form $i = m, 1 \le j \le n$. Trivially, the minimum total disruption a seam of length one can have is the disruption measure of the single pixel in the seam. Thus $T(1,j) = d(1,j)$, so our base case holds.

Now suppose the claim holds for some $1 < i < m$ and all arbitrary $1 \le j \le n$. Note that, starting from pixel $A(i-1,j)$, we will have (at most) three pixels to move to from $A(i-1,j)$. These pixels will be $A(i,j), A(i,j+1)$, and $A(i,j-1)$.[1]

Then, it must be the case that if we know the minimum total disruption of the seams starting at each of $A(i,j), A(i,j+1)$, and $A(i,j-1)$, we can simply take the minimum of those minima and add back the disruption $d(i-1,j)$ for the starting pixel $A(i-1,j)$. From the induction hypothesis, these minima will be held in $T(i,j), T(i,j+1)$, and $T(i,j-1)$, respectively. Thus it is the case that

$$T(i-1,j) = \min\{T(i,j), T(i,j+1), T(i,j-1)\} + d(i-1,j).$$

[1] If any of these entries do not exist (which is the case if $A(i-1,j)$ has either $j = 1$ or $j = n$), ignore those entries but maintain the others.

Now that we can be certain every entry $T(i, j)$ will correspond to the minimum total disruption of a seam starting at $A(i, j)$, we can easily return the minimum of the entries in the first row as the output of the algorithm, as these will indeed correspond to the minimum total disruptions of seams starting from each column in the first row of the image A.

Complexity. This algorithm has a runtime of $\mathcal{O}(mn)$. We iterate over every cell in T, which has dimensions $m \times n$, and each cell is filled in constant time, resulting in $\mathcal{O}(mn)$ total checks.