ECS 122A – Algorithm & Analysis
Homework 02 Solution

Prove or disapprove the time complexity guess for each of the following recurrences using the subsitution method.

1. $T(n) = T(n-1) + n$ is $O(n^2)$

   Answer:

   Proof:
   Let $k \geq 1$. Assume $T(n) \leq cn^2$ for some constant $c > 0$ and for all $1 \leq n < k$.

$$\begin{aligned} T(k) &= T(k-1) + k \\ &\leq c(k-1)^2 + k \\ &= ck^2 - 2ck + c + k \end{aligned}$$

   To show $T(k) \leq ck^2$, i.e.,

$$\begin{aligned} & ck^2 - 2ck + c + k \leq ck^2 \\ & -2ck + c + k \leq 0 \\ & c \leq (2c-1)k \\ & \frac{1}{k} \leq \frac{2c-1}{c} \text{ (both c and k are positive)} \\ & \frac{c}{2c-1} \leq k \end{aligned}$$

   Pick $c = 1$, then $k \geq 1$.
   So there exists $c > 0$ such that $T(k) \leq ck^2$.
   Conclusion: $T(n)$ is $O(n^2)$.

2. $T(n) = T(n/2) + 1$ is $O(logn)$

   Answer:

   Proof:
   Let $k \geq 1$. Assume $T(n) \leq clogn$ for some constant $c > 0$ and for all $1 \leq n < k$.

$$\begin{aligned} T(k) &= T(\frac{k}{2}) + 1 \\ &\leq clog\frac{k}{2} + 1 \\ &= clogk - clog2 + 1 \\ &= clogk - c + 1 \end{aligned}$$

To show $T(k) \leq clogk$, i.e.,

$$\begin{aligned} clogk - c + 1 &\leq clogk \\ c &\geq 1 \end{aligned}$$

Pick $c = 1$. So there exists $c > 0$ such that $T(k) \leq clogk$.
Conclusion: $T(n)$ is $O(logn)$

3. $T(n) = T(n/2) + n^2$ is $O(nlogn)$

   Answer:

   Intuitively, $T(n)$ has n^2 in it. It must be at least $O(n^2)$.

   Try to prove that $T(n)$ is $O(nlogn)$ using substitution:
   Let $k \geq 1$. Assume $T(n) \leq cnlogn$ for some constant $c > 0$ and for all $1 \leq n < k$.

$$\begin{aligned} T(k) &= T(\frac{k}{2}) + k^2 \\ &\leq c(\frac{k}{2})log\frac{k}{2} + k^2 \\ &= \frac{ck}{2}logk - \frac{ck}{2} + k^2 \end{aligned}$$

   To show $T(k) \leq cklogk$, i.e,.

$$\begin{aligned} \frac{ck}{2}logk - \frac{ck}{2} + k^2 &\leq cklogk \\ -\frac{c}{2} + k &\leq \frac{c}{2}logk \\ 2k &\leq clogk + c \end{aligned}$$

   When $k \geq 1$, $k > logk$. So no matter what c is, $2k$ will be larger than $clogk + c$ for sufficiently large k.

   If we try to add a lower order term such as dn: Assume $T(n) \leq cnlogn + dn$ for some constant $c > 0$ and for all $1 \leq n < k$.

$$\begin{aligned} T(k) &= T(\frac{k}{2}) + k^2 \\ &\leq c(\frac{k}{2})log\frac{k}{2} + d(\frac{k}{2}) + k^2 \\ &= \frac{ck}{2}logk - \frac{ck}{2} + \frac{dk}{2} + k^2 \end{aligned}$$

   To show $T(k) \leq cklogk + dk$, i.e.,

$$\begin{aligned} \frac{ck}{2}logk - \frac{ck}{2} + \frac{dk}{2} + k^2 &\leq cklogk + dk \\ -\frac{c}{2} + k &\leq \frac{c}{2}logk + \frac{d}{2} \\ 2k &\leq clogk + d + c \end{aligned}$$

   When $k \geq 1$, $k > logk$. So no matter what c and d are, $2k$ will be larger than $clogk + d + c$ for sufficiently large k.
   Conclusion: $T(n)$ is not $O(nlogn)$.

4. $T(n) = 3T(\frac{n}{2}) + n$ is $O(n^{log3})$

   Answer:

   Proof:
   Let $k \geq 1$. Assume $T(n) \leq cn^{log3}$ for some constant $c > 0$ and for all $1 \leq n < k$.

$$\begin{aligned} T(k) &= 3T(\frac{k}{2}) + k \\ &\leq 3c(\frac{k}{2})^{log3} + k \\ &= \frac{3c}{2^{log3}} k^{log3} + k \\ &= ck^{log3} + k \end{aligned}$$

   To show $T(k) \leq ck^{log3}$, i.e,.

$$\begin{aligned} ck^{log3} + k &\leq ck^{log3} \\ k &\leq 0 \end{aligned}$$

   which cannot be true.

   Try adding a lower order term dn:
   Let $k \geq 1$. Assume $T(n) \leq cn^{log3} + dn$ for some constants $c > 0$ and d and for all $1 \leq n < k$.

$$\begin{aligned} T(k) &= 3T(\frac{k}{2}) + k \\ &\leq 3(c(\frac{k}{2})^{log3} + d\frac{k}{2}) + k \\ &= ck^{log3} + (\frac{3d}{2} + 1)k \end{aligned}$$

   To show $T(k) \leq ck^{log3} + dk$, i.e,.

$$\begin{aligned} & ck^{log3} + (\frac{3d}{2} + 1)k \leq ck^{log3} + dk \\ & \frac{3d}{2} + 1 \leq d \\ & d \leq -2 \end{aligned}$$

   Pick $d = -2, c = 1$ (c can be any positive number). So there exists $c > 0$ such that $T(k) \leq ck^{log3} + dk$.
   Conclusion: $T(n)$ is $O(n^{log3})$.

5. $T(n) = T(n-1) + T(\frac{n}{2}) + n$ is $O(n2^n)$

   Answer:

   Proof:
   Let $k \geq 1$. Assume $T(n) \leq cn2^n$ for some constant $c > 0$ and for all $1 \leq n < k$.

$$\begin{aligned} T(k) &= T(k-1) + T(\frac{k}{2}) + k \\ &\leq c(k-1)2^{k-1} + \frac{ck}{2}2^{\frac{k}{2}} + k \end{aligned}$$

To show $T(k) \leq ck2^k$, i.e,.

$$\begin{aligned}
c(k-1)2^{k-1} + \frac{ck}{2}2^{\frac{k}{2}} + k &\leq ck2^k \\
\frac{c(k-1)}{2}2^k + \frac{ck}{2}2^{\frac{k}{2}} + k &\leq ck2^k \\
\frac{ck}{2}2^{\frac{k}{2}} + k &\leq \frac{c(k+1)}{2}2^k \\
ck2^{\frac{k}{2}} + 2k &\leq c(k+1)2^k \\
ck2^{\frac{k}{2}} + 2k &\leq ck2^k + c2^k
\end{aligned}$$

First, $ck2^{\frac{k}{2}} \leq ck2^k$ for any $c > 0$ when $k \geq 1$.
Second, $2k \leq c2^k$ when $c = 2$ and $k \geq 1$.
So pick $c = 2$, then $ck2^{\frac{k}{2}} + 2k \leq ck2^k + c2^k$.
Conclusion, $T(n)$ is $O(n2^n)$.