CS3230 Tutorial 4

AY 25/26 Sem 1 — github/omgeta

Q1). Given $A(10) = 352, B(10) = 221$:

$$
\begin{aligned}
c_0 &= a_0 b_0 = 2 \cdot 1 = 2 \\
c_1 &= a_1 b_0 + a_0 b_1 = 5 \cdot 1 + 2 \cdot 2 = 9 \\
c_2 &= a_2 b_0 + a_1 b_1 + a_0 b_2 = 3 \cdot 1 + 5 \cdot 2 + 2 \cdot 2 = 17 \\
c_3 &= a_2 b_1 + a1 b_2 = 3 \cdot 2 + 5 \cdot 2 = 16 \\
c_4 &= a2 b_2 = 3 \cdot 2 = 6
\end{aligned}
$$

Then $C(10) = 77792$

Q2). Given $A(10) = 10(3 \cdot 10 + 5) + 2 \implies A_1 = 35, A_2 = 2$,
$B(10) = 10(2 \cdot 10 + 2) + 1 \implies B_1 = 22, B_1 = 1$

$$
\begin{aligned}
A_1 B_1 &= \cdot 35 \cdot 22 = 770 \\
A_1 B_2 &= 35 \cdot 1 = 35 \\
A_2 B_1 &= 2 \cdot 22 = 44 \\
A_2 B_2 &= 2 \cdot 1 = 2
\end{aligned}
$$

Then $C(10) = 10^2(A_1 B_1) + 10(A_1 B_2 + A_2 B_1) + A_2 B_2 = 77792$

Q3). $T(n) = 4T(n/2) + \Theta(n)$ which by Master Theorem Case 1, $T(n) = \Theta(n^2)$

Q4). To speedup compute $(a + c)(c + d) = ac + ad + bc + bd$ then subtract ac, bd since we only need $ad + bc$. Then $T(n) = 3T(n/2) + \Theta(n) = \Theta(n^{1.58})$

Q5). For a sorted list, binary search for largest i s.t. $A[i] \geq i$ for $O(\log n)$.

Q6). Split into 3 groups and weight 2 each. If they are equal, recurse on the remaining group. If not, recurse on the larger group. At each step we divide problem size by 3, so for k weighings of x weights, we have $k \geq \lceil \log_3 x \rceil$