CSCI 3104: Algorithms
Spring 2022
Recitation #6 - Recursion

SOLUTIONS

Some Helpful Identities

$$a\sum_{i=0}^{n} r^i = a\left(\frac{1-r^{n+1}}{1-r}\right) \qquad \text{(Finite Geometric Series)}$$

$$a\sum_{i=0}^{n} i = \frac{an(n+1)}{2} \qquad \text{(Sum of first } n \text{ integers/special case of Finite Arithmetic Series)}$$

$$\sum_{i=0}^{n} ix^i = \frac{x(nx^{n+1}-(n+1)x^n+1)}{(x-1)^2}$$

$$\sum_{i=0}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$$

Problem 1

For the following algorithms, write the recurrence relation and the base cases. (*note:* we index sets starting from 1)

a. .

Algorithm 1 Recurrence 1.a

```
procedure Max1(Integer Array A)
    L = |A|

    if L = 1 then return A[1]
    else if L = 0 then return −∞

    m_1 ← Max1(A[1 : L/2])
    m_2 ← Max1(A[L/2 + 1 : L])
return m_1
```

First, note that this "max" function does not return the max but the first element in the array. It may also return $-\infty$ depending on how the division of lists is defined. Max1 has two base cases: if $|A| = 1$ or $|A| = 0$, in which case it simply returns a value. We say this call takes time $\Theta(1)$. Otherwise, it divides the input array into two sections and recursively computes each. Thus,

$$T(n) = \begin{cases} \Theta(1) & n \leq 1 \\ 2T(n/2) + \Theta(1) & n > 1 \end{cases}$$

BONUS: compute the runtime of this recursion.

We use the unrolling method:

1. *Determine the number of times to unroll.* For fixed input size n, we reach our base cases after $i = \log_2 n$ iterations: let i be the number of iterations we need to reach a base case. Since our input size is halved at each step, we have $n/2^i = 1$ and can solve for i.
2. *Write out several iterations.* We try a few to see a pattern:

$$\begin{aligned} T(n) &= 2T(n/2) + c \\ &= 2(2T(n/4) + c) + c \\ &= 8T(n/8) + 4c + 2c + c \end{aligned}$$

3. *Identify the pattern.* We see that the ith "unrolling" adds $2^i c$ to our runtime. Thus,

$$T(n) = \sum_{i=0}^{\log_2 n} c2^i$$

4. *Simplify.* We simplify using that $a \sum_{i=0}^{n} r^i = a\left(\frac{1-r^{n+1}}{1-r}\right)$:

$$T(n) = c\left(\frac{1 - 2^{\log_2 n+1}}{1-2}\right) = c(2n-1)$$

b. ˙

Algorithm 2 Recurrence 1.b

procedure MAX2(Integer Array A)
  $L = |A|$

  if $L = 1$ **then return** $A[1]$
  else if $L = 0$ **then return** $-\infty$

  $m \leftarrow$ MAX2($A[1 : L/2]$)

  for $i \leftarrow L$; $i \geq 1$; $i \leftarrow i - 1$ **do**
    if $m \geq A[i]$ **then** $m \leftarrow A[i]$
  return m

MAX2 has 2 base cases: when $|A| \leq 1$. In either of these cases, the computation is $\Theta(1)$. Otherwise, MAX2 makes one recursive call of size $n - 3$ and performs n computations. Thus,

$$T(n) = \begin{cases} \Theta(1) & n \leq 1 \\ T(n-3) + cn & n > 1 \end{cases}$$

BONUS: compute the runtime of this recursion.

We use the unrolling method:

1. *Determine the number of times to unroll.* For fixed input size n, we reach our base cases after $i = \frac{n-1}{3}$ iterations: let i be the number of iterations we need to reach a base case. Since our input size is reduced by 3 at each step, we have $n - 3i \leq 1$ and can solve for i.
2. *Write out several iterations.* We try a few to see a pattern:

$$\begin{aligned} T(n) &= T(n-3) + n \\ &= T(n-6) + n - 3 + n \\ &= T(n-9) + n - 6 + n - 3 + n \end{aligned}$$

3. *Identify the pattern.* We see that the ith "unrolling" adds $n - 3i$ to our computation:

$$T(n) = \sum_{i=0}^{\frac{n-1}{3}} (n - 3i)$$

4. *Simplify.* We simplify using that $a\sum_{i=0}^{n} i = \frac{an(n+1)}{2}$:

$$\begin{aligned} T(n) &= n\frac{(n-1)}{3} + 3\sum_{i=1}^{\frac{n-1}{3}} i \\ &= \frac{n(n-1)}{3} + \frac{3}{2}\left(\frac{n-1}{3}\right)\left(\frac{n+2}{3}\right) \\ &= \frac{(n-1)(3n+2)}{6} \end{aligned}$$

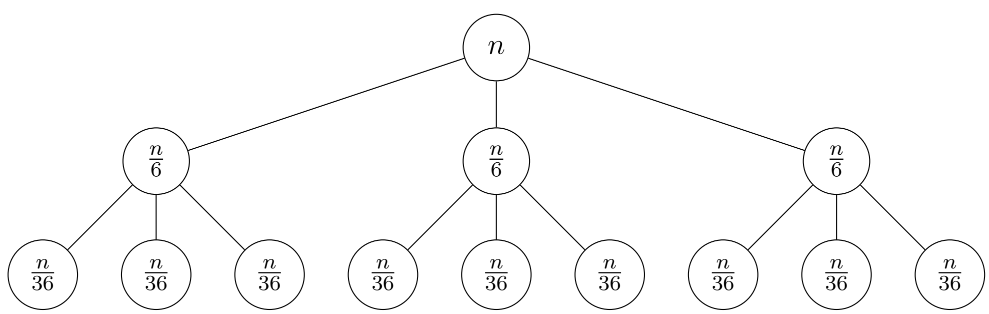
Problem 2

For each of the following recurrence relationships, use the **tree method** to obtain a closed-form expression for the runtime by doing the following: 1) draw a tree diagram of the first few layers of the function's recursive calls, 2) determine the depth of your tree, 3) sum work over all vertices, and 4) simplify.

a.

$$T(n) = \begin{cases} 3T(n/6) + n^2 & n > 3 \\ n^2 & n \leq 3 \end{cases}$$

1. *Draw the tree.*



2. *Determine the depth.* We reach our base case at the dth layer, when $n/6^d \leq 3$. Thus, we have depth $d \geq \log_6 n - \log_6 3$.
3. *Sum the work.* At layer i, there are 3^i vertices, each of which takes time $(n/6^i)^2 = n^2/6^{2i}$. Since we do the same work in the base cases as everywhere else, we roll them into the sum. Thus,

$$T(n) = \sum_{i=0}^{\log_6 n - \log_6 3} 3^i n^2/6^{2i}$$

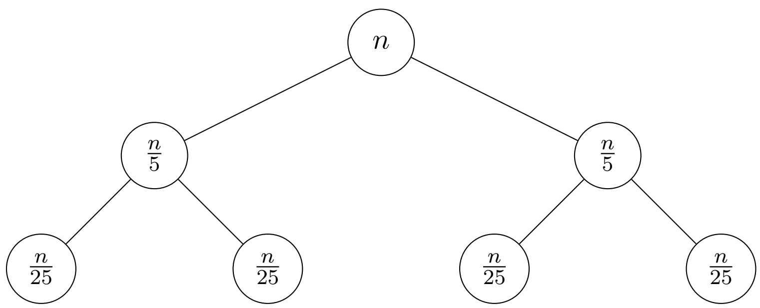
4. *Simplify.* We simplify this expression:

$$\begin{aligned}
T(n) &= n^2 \sum_{i=0}^{\log_6 n - \log_6 3} \left(\frac{1}{12}\right)^i \\
&= n^2 \left(\frac{1 - 12^{-\log_6 n + \log_6 3 - 1}}{1 - 12^{-1}}\right) \\
&= \frac{12}{11} n^2 \left(1 - 12^{-\log_6 n} 12^{\log_6 3 - 1}\right) \\
&= \frac{12}{11} n^2 \left(1 - \frac{12^{\log_6 3 - 1}}{12^{\frac{\log_{12} n}{\log_{12} 6}}}\right) = \frac{12}{11} n^2 \left(1 - \frac{12^{\log_6 3 - 1}}{n^{\frac{1}{\log_{12} 6}}}\right)
\end{aligned}$$

b.

$$T(n) = \begin{cases} 2T(n/5) + n \log_5 n & n > 1 \\ 2 & n \leq 1 \end{cases}$$

1. *Draw the tree.*



2. *Determine the depth.* We reach a base case and "bottom out" at depth $d = \log_5 n$. This can be seen by solving $n/5^d \leq 1$, when our input size is smaller than our base case, for d.
3. *Sum the work.* The ith layer has 2^i vertices, with each one running in time $\log_5(n/5^i) = \log_5 n - i$. Thus,

$$T(n) = \sum_{i=0}^{\log_5(n)} 2^i (\log_5 n - i) + 2^{\log_5(n)} 2$$

4. *Simplify.* We use the identity for $\sum_{i=0}^{n} i2^i$:

$$
\begin{aligned}
T(n) &= \log_5 n \sum_{i=0}^{\log_5 n} 2^i - \sum_{i=0}^{\log_5 n} (2^i i) + 2n^{\frac{1}{\log_2 5}} \\
&= \log_5 n \frac{1 - 2^{\log_5 n}}{1-2} - \sum_{i=0}^{\log_5 n} (2^i i) \\
&= n^{1/\log_2 5} \log_5 n + 2^{\log_5 n + 1}(\log_5 n - 1) + 2 \\
&= 3n^{1/\log_2 5} \log_5 n - n^{1/\log_2 5} + 2
\end{aligned}
$$

Problem 3

For each of the following recurrence relationships, use the **unrolling method** to obtain a closed-form expression for the runtime by doing the following: 1) determine the number of times to unroll, 2) write out several iterations, 3) identify the pattern, and 4) simplify.

a.

$$
T(n) = \begin{cases} 3T(n-4) + 2n & n > 4 \\ 3 & n \leq 4 \end{cases}
$$

1. *Determine the number of times to unroll.* Each iteration reduces the input size by 4, so we solve $n - 4d \leq 4$ for d. We unroll $d = (n-4)/4$ times.
2. *Write out several iterations.*

$$
\begin{aligned}
T(n) &= 3T(n-4) + 2n \\
&= 9T(n-8) + 3 \cdot 2(n-4) + 2n \\
&= 27T(n-12) + 9 \cdot 2(n-8) + 3 \cdot 2(n-4) + 2n
\end{aligned}
$$

3. *Identify the pattern.* At the ith level of unrolling, we add $3^i 2(n-4i)$ to our runtme.

$$
T(n) = \sum_{i=0}^{(n-4)/4} 3^i 2(n-4i)
$$

4. *Simplify.* Use the identity $\sum_{i=0}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$:

$$\begin{aligned}
T(n) &= \sum_{i=0}^{(n-4)/4} 3^i 2(n-4i) \\
&= 2n \sum_{i=0}^{(n-4)/4} 3^i - 12 \sum_{i=0}^{(n-4)/4} i^2 \\
&= 2n \frac{(1-3^{(n-4)/4})}{1-3} - 2\left(\left(\frac{(n-4)}{4}\right)\left(\frac{(n-4)}{4}+1\right)\left(\frac{(n-4)}{2}+1\right)\right) \\
&= n\left(3^{(n-4)/4}-1\right) - \left(\frac{(n-4)}{2}\right)\left(\frac{n}{4}\right)\left(\frac{(n-2)}{2}\right) \\
&= n\left(3^{(n-4)/4}-1\right) - \frac{n(n-2)(n-4)}{16}
\end{aligned}$$

b.

$$T(n) = \begin{cases} 4T(n-5) + n/4 & n > 6 \\ 3 & n \le 6 \end{cases}$$

1. *Determine the number of times to unroll.* We unroll d times, until $n - 5d \le 6$. Solving for d, we get $d = (n-6)/5$.
2. *Write out several iterations.*

$$\begin{aligned}
T(n) &= 4T(n-5) + n/4 \\
&= 16T(n-10) + 4(n-5)/4 + n/4 \\
&= 64T(n-15) + 16(n-10)/4 + 4(n-5)/4 + n/4
\end{aligned}$$

3. *Identify the pattern.* On the ith layer, we add $4^i(n-5i)/4$ to our runtime:

$$T(n) = \sum_{n=0}^{(n-6)/5} 4^i(n-5i)/4$$

4. *Simplify.*

$$\begin{aligned}
T(n) &= \sum_{n=0}^{(n-6)/5} 4^i(n-5i)/4 \\
&= \sum_{n=0}^{(n-6)/5} 4^i(n-5i)/4 \\
&= n \sum_{n=0}^{(n-6)/5} 4^i - \frac{5}{4} \sum_{n=0}^{(n-6)/5} i4^i
\end{aligned}$$

Use the identity $\sum_{i=0}^{n} ix^i = \frac{x(nx^{n+1}-(n+1)x^n+1)}{(x-1)^2}$:

$$
\begin{aligned}
T(n) &= n\frac{1-4^{(n-6)/5}}{1-4} - \frac{5}{(4-1)^2}\left(\frac{(n-6)}{5}4^{(n-6)/5+1} - \frac{(n-1)}{5}4^{(n-6)/5} + 1\right) \\
&= n\frac{4^{(n-6)/5}-1}{3} - \left((n-6)4^{(n-6)/5+1} - (n-1)4^{(n-6)/5} + \frac{1}{9}\right) \\
&= n\frac{4^{(n-6)/5}-1}{3} - \left((n-6)4^{(n-6)/5+1} - (n-1)4^{(n-6)/5} + \frac{1}{9}\right)
\end{aligned}
$$