Homework 2

October 9, 2025

1 [BvG 3.1]

Show that every 2-tree with n internal nodes has $n + 1$ external nodes.

Answer:

Here $f(n)$ be the number of external nodes in 2-tree with n internal nodes.

Let's start with Base case n = 0: $f(n) = n + 1 = 0 + 1 = 1$.

2-tree with 0 internal node means it only has root node, which is an external node too. Therefore, base case is true.

Inductive Hypothesis: Now let's assume it holds true for k such taht : $0 \leq k < n$

$f(k) = k + 1$

Suppose we have a 2-tree with $n \geq 1$ internal nodes. So the root node will have two children, i.e left subtree and right sub tree.

left subtree = T_L and right subtree = T_R

If k = internal nodes in T_L, then $n - k - 1$ = internal nodes in T_R

So: $f(n) = f(k) + f(n - k - 1)$

$f(n) = k + 1 + n - k - 1 + 1$

$f(n) = n + 1$

Outcomes: CS 6, 5870-1

2 [BvG 3.3]

Show that the external path length epl in a 2-tree with m external nodes satisfies $epl \leq \frac{1}{2}(m^2 + m - 2)$. Conclude that $epl \leq \frac{1}{2}n(n+3)$ for a 2-tree with n internal nodes.

Answer:

m = external nodes

n = internal nodes

So in 2-tree: m = n + 1

$m = m_1 + m_2$: 2-tree T has two subtrees T_1 and T_2.

T_1 has m_1 external node and T_2 has m_2 external node.

$epl \leq \frac{1}{2}(m^2 + m - 2)$

$epl_1 \leq \frac{1}{2}(m_1^2 + m_1 - 2)$

$epl_2 \leq \frac{1}{2}(m_2^2 + m_2 - 2)$

$epl_1 + epl_2 + m_1 + m_2 = epl \leq \frac{1}{2}((m_1 + m_2)^2 + (m_1 + m_2) - 2)$

$epl \leq \frac{1}{2}(m_1^2 + m_1 - 2) + \frac{1}{2}(m_2^2 + m_2 - 2) + m_1 + m_2 \leq \frac{1}{2}((m_1 + m_2)^2 + (m_1 + m_2) - 2)$

$epl \leq \frac{1}{2}(m_1^2 + m_1) + \frac{1}{2}(m_1 + m_2) - \frac{2}{2} - \frac{2}{2} + m_1 + m_2 \leq \frac{1}{2}(m_1^2 + m_2^2) + m_1 m_2 + \frac{1}{2}(m_1 + m_2) - 1$

$epl \leq \frac{1}{2}(m_1^2 + m_2^2) + \frac{3}{2}(m_1 + m_2) - 2 \leq \frac{1}{2}(m_1^2 + m_2^2) + m_1 m_2 + \frac{1}{2}(m_1 + m_2) - 1$

$epl \leq \frac{3}{2}(m_1 + m_2) - 2 \leq m_1 m_2 + \frac{1}{2}(m_1 + m_2) - 1$

$epl \leq \frac{3}{2}(m_1 + m_2) - \frac{1}{2}(m_1 + m_2) - 2 \leq m_1 m_2 - 1$

$epl \leq m_1 + m_2 - 2 \leq m_1 m_2 - 1$

$epl \leq 0 \leq m_1 m_2 - m_1 - m_2 - 1$

$epl \leq 0 \leq (m_1 - 1)(m_2 - 1)$

To show: $epl \leq \frac{1}{2}n(n+3)$

Substitute: m = n + 1

$epl \leq \frac{1}{2}((n+1)^2 + n + 1 - 2)$

$epl \leq \frac{1}{2}(n^2 + 2n + 1 + n + 1 - 2)$

$epl \leq \frac{1}{2}(n^2 + 3n)$

$epl \leq \frac{1}{2}n(n+3)$

Outcomes: CS 6, 5870-1

3 [SW 2.1.8]

Suppose that we use insertion sort on a randomly ordered array where elements have only one of three values. Is the running time linear, quadratic, or something in between?

Answer:

Worst Case: $O(n^2)$.

Insertion sort might have to go through the items in the array and then shift n times if the array is in descending order.

However, since we only have 3 distinct randomly ordered array, the running time could be in between because we might not have to do as many shifts we normally would do on all distinct items.

Outcomes: CS 6, 5870-1

4 [4.5-1]

Use the master method to give tight asymptotic bounds for the following recurrences.

a $T(n) = 2T(n/4) + 1$

b $T(n) = 2T(n/4) + \sqrt{n}$

c $T(n) = 2T(n/4) + n$

d $T(n) = 2T(n/4) + n^2$

Answer:

a. $T(n) = 2T(n/4) + 1$

Here: a $= 2$, b $= 4$, f(n) $= 1$

So $lg_b a = lg_4 2 = \frac{1}{2}$

Therefore, Case 1 says: $f(n) = 1$ is samller than $n^{\frac{1}{2}}$

$T(n) = \Theta(n^{\frac{1}{2}})$

b. $T(n) = 2T(n/4) + \sqrt{n}$

Here $a = 2, b = 4, f(n) = \sqrt{n}$

So $lg_b a = lg_4 2 = \frac{1}{2}$

Therefore, Case 2 says: $f(n) = \sqrt{n}$ is similar to $n^{\frac{1}{2}}$

$T(n) = \Theta(\sqrt{n}.lgn)$

c. $T(n) = 2T(n/4) + n$

Here $a = 2, b = 4, f(n) = n$

So $lg_b a = lg_4 2 = \frac{1}{2}$

Therefore, Case 3 says: $f(n) = n$ is larger than $n^{\frac{1}{2}}$

$T(n) = \Theta(n)$

d. $T(n) = 2T(n/4) + n^2$

Here $a = 2, b = 4, f(n) = n^2$

So $lg_b a = lg_4 2 = \frac{1}{2}$

Therefore, Case 3 says: $f(n) = n^2$ is larger than $n^{\frac{1}{2}}$

$T(n) = \Theta(n^2)$

Outcomes: CS 6, 5870-1

5 [CLRS Problem 4-3 (modified)]

Give an asymptotically tight bound for $T(n)$ in each of the following recurrences.

a $T(n) = 4T(n/3) + n \lg n$

b $T(n) = 4T(n/2) + n^2\sqrt{n}$

c $T(n) = T(n/2) + T(n/4) + T(n/8) + n$

d $T(n) = T(n-1) + 1/n$

e $T(n) = T(n-1) + \lg n$

Answer:

a. $T(n) = 4T(n/3) + n \lg n$

Here: a = 4, b = 3, f(n) = n lg n

So $lg_b a = lg_3 4 \approx 1.26$

Therefore, Case 1 says: $f(n) = nlgn$ is smaller than $n^{1.26}$

$T(n) = \Theta(n^{1.26})$

b. $T(n) = 4T(n/2) + n^2\sqrt{n}$

Here: a = 4, b = 2, $f(n) = n^2\sqrt{n}$

So $lg_b a = lg_2 4 = 2$

Therefore, Case 3 says: $f(n) = n^2\sqrt{n}$ is greater than n^2

$T(n) = \Theta(n^2\sqrt{n})$

c. $T(n) = T(n/2) + T(n/4) + T(n/8) + n$

Using recurrence tree:

$T(n) = n + T(n/2) + T(n/4) + T(n/8)$

$$
\begin{aligned}
T(n) &= n + T\left(\frac{n}{2}\right) + T\left(\frac{n}{4}\right) + T\left(\frac{n}{8}\right) \\
&= n + \left[\frac{n}{2} + T\left(\frac{n}{4}\right) + T\left(\frac{n}{8}\right) + T\left(\frac{n}{16}\right)\right] \\
&\quad + \left[\frac{n}{4} + T\left(\frac{n}{8}\right) + T\left(\frac{n}{16}\right) + T\left(\frac{n}{32}\right)\right] \\
&\quad + \left[\frac{n}{8} + T\left(\frac{n}{16}\right) + T\left(\frac{n}{32}\right) + T\left(\frac{n}{64}\right)\right] \\
&\quad \vdots
\end{aligned}
$$

At each level, the total work is a decreasing geometric sum:

$$\text{Level 0: } n, \quad \text{Level 1: } \frac{n}{2}+\frac{n}{4}+\frac{n}{8}=\frac{7n}{8}, \quad \text{Level 2: } \left(\frac{7}{8}\right)^2 n, \quad \dots$$

Total work across all levels:

$$T(n)=n\left(1+\frac{7}{8}+\left(\frac{7}{8}\right)^2+\cdots\right)=n\cdot\sum_{i=0}^{\infty}\left(\frac{7}{8}\right)^i=n\cdot\frac{1}{1-\frac{7}{8}}=8n$$

$T(n)=\Theta(n)$

d. $T(n)=T\left(n-1\right)+1/n$

Using recurrence tree:

$T(n)=1/n+T\left(n-1\right)$

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

$T(n)=\frac{1}{n-2}+T\left(n-3\right)$

$T(n)=\frac{1}{n-3}+T\left(n-4\right)$

Continue....

$T(2)=\frac{1}{2}+T\left(1\right)$

$T(1)=\frac{1}{1}+T\left(0\right)$

We can see a pattern here: $T(n=)\frac{1}{1}+\frac{1}{2}+\frac{1}{3}\cdots+\frac{1}{n}$

This is harmonic series. We know that Harmonic Series is : $O(\lg n)$

Therefore, $T(n)=\Theta(\lg n)$

e. $T(n)=T\left(n-1\right)+\lg n$

Using recurrence tree:

$T(n)=\lg n+T(n-1)$

$T(n-1)=\lg n-1+T(n-2)$

$T(n-2)=\lg n-2+T(n-3)$

$T(n-3)=\lg n-3+T(n-4)$

Continue .....

$T(2)=\lg 2+T(1)$

$T(1)=\lg 1+T(0)$

We can see a pattern here: $T(n)=\lg n+\lg n-1+\lg n-2+\cdots+\lg 2+\lg 1.$

$T(n)=\lg[n*(n-1)*(n-2)*\cdots*2*1].$

Which is $\lg n!$. We know that is: $\Theta(n\lg n)$

Outcomes: CS 6, 5870-1

6 [GT C-5.2]

Given an array A of elements that come from a total order, an *inversion* in A is a pair of elements that are in the wrong order, that is, a pair (i, j) where $i < j$ and $A[i] > A[j]$. Show that the in-place version of insertion sort, as given in Algorithm 5.5, runs in $O(n+I)$ time, where I is the number of inversions in the array A.

Answer:

The outer for loop runs: n - 1 times (from $i < -1$ to $n - 1$).

So we can say that total outer loop's work: O(n)

The inner while loop runs i times. On worst case we do i shifts.

Each shift we do, it takes care of one inversion. Therefore, we can say that total shifts = total Inversions = I

One shift takes O(1) time.

Since shift = Inversion = I, then we can say that total times = O(I) for I inversions.

We can combine the work:

T(n) = O(n) + O(I)

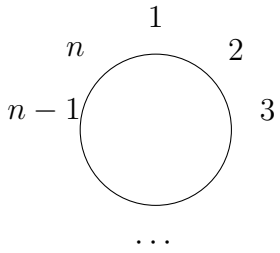
T(n) = O(n + I)

Outcomes: CS 2, CS 6, 5870-1

7 [The Josephus Problem]

Suppose we have n items arranged in a circle as shown:

Figure 1: Numbered items arranged in a circle



We proceed around the circle, removing every other item (item 2 is the first to be removed) until one item remains. For example, if $n = 10$, the sequence of removed items is $2, 4, 6, 8, 10, 3, 7, 1, 9$, with item 5 being the last one left. The *survivor's number*, $J(n)$, is the number of the last remaining item from a set of n items (thus, $J(10) = 5$).

a. Suppose there are $2n$ items. After n items have been removed, there are n items remaining. What are the numbers of the items that remain? How do those numbers relate to the numbering used for an initial set of n? Use this information to express $J(2n)$ in terms of $J(n)$.

b. Suppose there are now $2n+1$ items. After $n+1$ items have been removed, there are n items remaining. What are the numbers of the items that remain? How do those numbers relate to the numbering used for an initial set of n? Use this information to express $J(2n+1)$ in terms of $J(n)$.

c. Combine parts **a** and **b**, along with the base case $J(1) = 1$, to form a recurrence relation for $J(n)$.

d. Use the recurrence relation to create a small table of $J(n)$ values (you shouldn't need more than 20 to see the pattern). Use this table to find a closed form (*i.e.*, non-recursive) for $J(n)$. *Hint*: Express n as $n = 2^m + l$, where $2^m \leq n < 2^{m+1}$.

e. Prove that your closed form solution is correct. *Hint*: Use induction on m.

Answer:

a. We are left with odd numners. J(2n) = 2J(n) - 1

b. We are left with even numbers. J(2n+1) = 2J(n) + 1

c. Combining a and b:

$$J(n) = \begin{cases} J(1) = 1 & \text{base case} \\ 2J(\frac{n}{2}) - 1 & \text{if } n \text{ is even} \\ 2J(\lfloor \frac{n}{2} \rfloor) + 1 & \text{if } n \text{ is odd and } n > 1 \end{cases}$$

d. We found from the table that if n is power of 2 then our J(n) is always 1.

We will use hint for other number.
Hint: Express n as $n = 2^m + l$, where $2^m \leq n < 2^{m+1}$.

$l = n - 2^m$

Our closed form: $J(n) = 2l + 1$ where $n = 2^m + l$ and $0 \leq l < 2^m$

e. To show closed form solution is correct:

Base Case: m = 0
Then, $2^0 = 1$
So l = 0
Inductive Hypothesis:
Outcomes: CS 6, 5870-1, 5870-2

8 [CLRS Problem 4-4]

This problem develops properties of the Fibonacci numbers, which are defined by recurrence (3.22). We shall use the technique of generating functions to solve the Fibonacci recurrence. Define the *generating function* (or *formal power series*) $\mathscr{F}$ as

$$\begin{aligned}\mathscr{F}(z) &= \sum_{i=0}^{\infty} F_i z^i \\ &= 0 + z + z^2 + 2z^3 + 3z^4 + 5z^5 + 8z^6 + 13z^7 + 21z^8 + \cdots\end{aligned}$$

where F_i is the i^{th} Fibonacci number.

a Show that $\mathscr{F}(z) = z + z\mathscr{F}(z) + z^2\mathscr{F}(z)$.

b Show that

$$\begin{aligned}\mathscr{F}(z) &= \frac{z}{1 - z - z^2} \\ &= \frac{z}{(1 - \phi z)\left(1 - \hat{\phi} z\right)} \\ &= \frac{1}{\sqrt{5}}\left(\frac{1}{1 - \phi z} - \frac{1}{1 - \hat{\phi} z}\right)\end{aligned}$$

where

$$\phi = \frac{1 + \sqrt{5}}{2} \approx 1.61803\ldots$$

and

$$\hat{\phi} = \frac{1 - \sqrt{5}}{2} \approx -0.61803\ldots$$

c Show that

$$\mathscr{F}(z) = \sum_{i=0}^{\infty} \frac{1}{\sqrt{5}}\left(\phi^i - \hat{\phi}^i\right) z^i$$

d Use part (c) to prove that $F_i = \phi^i/\sqrt{5}$ for $i \geq 0$, rounded to the nearest integer. *Hint*: Observe that $\left|\hat{\phi}\right| < 1$.

Answer:

a. We know that $F_0 = 0$ and $F_1 = 1$

$$\mathscr{F}(z) = \sum_{i=0}^{\infty} F_i z^i = F_0 z^0 + F_1 z^1 + \sum_{i=2}^{\infty} F_i z^i$$

$$\mathscr{F}(z) = 0 + z + \sum_{i=2}^{\infty} F_i z^i = z + \sum_{i=2}^{\infty} F_i z^i$$

$$We know : F_i = F_{i-1} + F_{i-2}$$

$$\mathscr{F}(z) = z + \sum_{i=2}^{\infty} (F_{i-1} + F_{i-2}) z^i$$

$$\mathscr{F}(z) = z + \sum_{i=2}^{\infty} F_{i-1} z^i + \sum_{i=2}^{\infty} F_{i-2} z^i$$

$$Let : k = i - 1 : \mathscr{F}(z) = \sum_{i=2}^{\infty} F_{i-1} z^i = z \sum_{k=1}^{\infty} F_k z^k$$

$$F_0 z^0 = 0, start : k = 0 : \mathscr{F}(z) = z \sum_{k=0}^{\infty} F_k z^k = zF(z)$$

$$Let : k = i - 2, start : k = 0 : \mathscr{F}(z) = \sum_{k=0}^{\infty} F_k z^{k+2} = z^2 \sum_{k=0}^{\infty} F_k z^k$$

$$\mathscr{F}(z) = z^2 F(z)$$

$$Combine : \mathscr{F}(z) = z + zF(z) + z^2 F(z)$$

b. To show:

$$\mathscr{F}(z) = \frac{z}{1 - z - z^2}$$

$$Given : \mathscr{F}(z) = z + zF(z) + z^2F(z)$$
$$\mathscr{F}(z) - zF(z) - z^2F(z) = z$$

$$\mathscr{F}(z)(1 - z - z^2) = z$$

$$\mathscr{F}(z) = \frac{z}{(1 - z - z^2)}$$

$$\mathscr{F}(z) = \frac{z}{(1 - \phi z)\left(1 - \hat{\phi} z\right)}$$

$$\mathscr{F}(z) = \frac{z}{(1 - \phi z)\left(1 - \hat{\phi} z\right)} = \frac{a}{(1 - \phi z)} + \frac{b}{\left(1 - \hat{\phi} z\right)}$$

$a = 1/\sqrt{5}, b = -1/\sqrt{5}$

$$\mathscr{F}(z) = \frac{1}{\sqrt{5}}\left(\frac{1}{(1 - \phi z)} - \frac{1}{\left(1 - \hat{\phi} z\right)}\right)$$

c. We have:

$$\mathscr{F}(z) = \frac{1}{\sqrt{5}}\left(\frac{1}{(1 - \phi z)} - \frac{1}{\left(1 - \hat{\phi} z\right)}\right)$$

Use geometric series to both fractions:

$$i- > \frac{1}{1-\phi z} = \sum_{i=0}^{\infty}(\phi z)^i = \sum_{i=0}^{\infty}\phi^i z^i$$

$$ii- > \frac{1}{1-\hat{\phi} z} = \sum_{i=0}^{\infty}(\hat{\phi} z)^i = \sum_{i=0}^{\infty}\hat{\phi}^i z^i$$

$$Substitute: i, and, ii: \mathscr{F}(z) = \frac{1}{\sqrt{5}}\left(\sum_{i=0}^{\infty}\phi^i z^i - \sum_{i=0}^{\infty}\hat{\phi}^i z^i\right)$$

$$\mathscr{F}(z) = \sum_{i=0}^{\infty}\frac{1}{\sqrt{5}}(\phi^i - \hat{\phi}^i)z^i$$

d.
From c, we can get that the coefficient of z^i that is $F_i = \frac{\phi^i - \hat{\phi}^i}{\sqrt{5}}$

Hint: $\left|\hat{\phi}\right| < 1$, $\hat{\phi}$ approaches 0 as i increases.

Therefore for $i \geq 0$, F_i is extremely close to the value of first term:$F_i \approx \frac{\phi^i}{\sqrt{5}}$

Since the second term: $\frac{-\hat{\phi}^i}{\sqrt{5}}$ is a very small number.

$F_i = \frac{\phi^i}{\sqrt{5}}$

Outcomes: CS 6, 5870-1

9 [GT A - 11.6]

Suppose you have a geometric description of the buildings of Manhattan and you would like to build a representation of the New York skyline. That is, suppose you are given a description of a set of rectangles, all of which have one of their sides on the x-axis, and you would like to build a representation of the union of all these rectangles. Formally, since each rectangle has a side on the x-axis, you can assume that you are given a set, $S = \{[a_1, b_1], [a_2, b_2], \ldots, [a_n, b_n]\}$ of sub-intervals in the interval $[0, 1]$, with $0 \leq a_i < b_i \leq 1$ for $i = 1, 2, \ldots, n$, such that there is an associated height h_i for each interval $[a_i, b_i]$ in S. The *skyline* of S is defined to be a list of pairs $[(x_0, c_0), (x_1, c_1), (x_2, c_2), \ldots, (x_m, c_m), (x_{m+1}, 0)]$, with $x_0 = 0$ and $x_{m+1} = 1$, and ordered by x_i values, such that each subinterval $[x_i, x_{i+1}]$ is the maximal subinterval that has a single highest interval, which is at height c_i, in S, containing $[x_i, x_{i+1}]$, for $i = 0, 1, \ldots, m$. Design a $\Theta(n \log n)$-time algorithm for computing the skyline of S.

Algorithm 1 Compute the Skyline of S

Input: A set $S = \{[a_1, b_1, h_1], [a_2, b_2, h_2], \ldots, [a_n, b_n, h_n]\}$ of rectangles, where $[a_i, b_i]$ is the horizontal interval and h_i is the height.
Output: The skyline list of key points $answer = [(x_0, c_0), (x_1, c_1), \ldots, (x_m, c_m), (x_{m+1}, 0)]$
edges ← [ ]
for each rectangle i in S **do**
    append (a_i, i) to edges
    append (b_i, i) to edges
end for
sort edges by non-decreasing x-coordinate
live ← [ ] ▷ Priority queue of active rectangles storing $(-h_i, b_i)$
answer ← [ ]
idx ← 0
while idx < len(edges) **do**
    curr_x ← edges[idx][0]
    while idx < len(edges) **and** edges[idx][0] = curr_x **do**
        b ← edges[idx][1]
        if $a_b = curr\_x$ **then**
            push $(-h_b, b_b)$ into live
        end if
        while live not empty **and** live[0][1] ≤ curr_x **do**
            pop from live
        end while
        idx ← idx +1
    end while
    max_height ← −live[0][0] if live not empty else 0
    if answer is empty **or** max_height ≠ answer[-1][1] **then**
        append $(curr\_x, max\_height)$ to answer
    end if
end while
return answer

Outcomes: CS 6, 5870-1

10 [GT C - 11.3]

There is a sorting algorithm, "Stooge-sort," which is named after the comedy team, "The Three Stooges." If the input size, n, is 1 or 2, then the algorithm sorts the input immediately. Otherwise, it recursively sorts the first $2n/3$ elements, then the last $2n/3$ elements, and then the first $2n/3$ elements again. The details are shown in the following algorithm. Show that Stooge-sort is correct and characterize the running time, $T(n)$, for Stooge-sort, using a recurrence equation, and use the master theorem to determine an asymptotic bound for $T(n)$.

Algorithm 2 Stooge Sort

```
1:  procedure StoogeSort(Comparable A[], int low, int high)
2:      n ← high − low + 1                          ▷ Determine size of list
3:      if n = 2 then
4:          if A[low] > A[high] then
5:              Swap(A[low], A[high])
6:          end if
7:      else if n > 2 then
8:          m ← ⌊n/3⌋                               ▷ Calculate overlap
9:          StoogeSort(A, low, high − m)            ▷ Recursively sort first part
10:         StoogeSort(A, low + m, high)            ▷ Recursively sort second part
11:         StoogeSort(A, low, high − m)            ▷ Recursively sort first part again
12:     end if
13: end procedure
```

Answer:

a. Show that Stooge-sort is correct

If $n < 2$, then do nothing. That is , if there is only one item or no item the list/array is sorted.

Base Case: $n = 2$:
it compares and swap elements if needed.

Thefore, the subarray is sorted.

Inductive Hypothesis:
Assume that for all $k < n$, this algorithm holds true.

For $n > 2$:

Let's divide the array into 3 sections: A, B and C.

i. First Recursive Call: StoogeSort(A[1...m])
A and B are sorted [Structural Induction]
We can say that everything in B is larger than in A.

ii. Second Recursive Call: StoogeSort(A[n-m+1...n])
This makes sure that largest elements are moved to C section.

Everythign in C is larger than everything in A and B
There could have been small elements in part C that needed to over A and B and not there yet.

iii. Third Recursive Call: StoogeSort(A[1...m])
This sort the elements in A and B again. The A and B contains smallest m elements, sorting this puts elements in correct position
Since the second recursive call makes sure that C has the largest elements and are sorted correctly.
Then final recursive call makes sure that the elements in A and B are sorted. We can say that Stooge-sort is correct.

b. Determine an asymptotic bount for T(n)

$T(n) = 3T(\frac{2}{3}n) + 0$

So, $a = 3, b = \frac{3}{2}, F(n) = 0, lg_b a \approx 2.71$

As per M.T Case 1, $F(n) = 0$ is smaller than $O(n^{2.71})$

Therefore, $T(n) = O(n^{2.71})$

Outcomes: CS 6, 5870-1