Lecture 3:
Master Theorem &
Proof of Correctness by Induction

Last lecture

We discussed

- how to use recurrence relation to express the run time of recursive algorithms
- How to solve recurrence relation using telescoping and recursion tree

Today’s lecture

- Master theorem for solving recurrence relation
- Prove correctness of a recursive algorithm: induction

A general recurrence relation

$T(n) = aT(\frac{n}{b}) + cn^d$ for some constants a, b, c, d

Let's use recursion tree to figure its run time:

$$T(n) = \left(1 + \left(\frac{a}{b^d}\right)^2 + \cdots + \left(\frac{a}{b^d}\right)^k\right) cn^d$$

- If $\frac{a}{b^d} < 1, T(n) = O\left(n^d\right)$
- If $\frac{a}{b^d} = 1, T(n) = O\left(n^d \log n\right)$
- If $\frac{a}{b^d} > 1, T(n) = O\left(\left(\frac{a}{b^d}\right)^k cn^d\right) = O(n^{\log_b a})$

$$\left(\frac{a}{b^d}\right)^{\log_b n} = n^{\log_b\left(\frac{a}{b^d}\right)} = n^{\log_b a - \log_b b^d}$$
$$= n^{\log_b a - \mathrm{d}}$$

Master Theorem

$$T(n) = aT(\frac{n}{b}) + cn^d$$

Case 1: $a < \mathrm{b}^{\mathrm{d}}$, or equiv. $d > \log_b a$, $T(n) = O\left(n^d\right)$

Case 2: $a = b^d$, or equiv. $d = \log_b a$, $T(n) = O\left(n^d \log n\right)$

Case 3: $a > b^d$, or equiv. $d < \log_b a$, $T(n) = O\left(n^{\log_b a}\right)$

Limit of master theorem

- Can only be used to solve the recurrence relation of specific form

$$T(n) = aT(\frac{n}{b}) + cn^d$$

- For other forms, for example:
  - $T(n) = T(n-1) + c$
  - $T\ (n) = T\left(\frac{4}{5}n\right) + 2T\left(\frac{2}{3}n\right) + cn$

Telescoping or recursion tree can be used

How do we prove the correctness of recursive algorithms?

- Proof by induction

Correctness of Merge sort

- Base case: input size n = 1, it is already sorted, Merge_sort correctly outputs sorted array
- Inductive assumption: Assume that Merge_sort correctly sorts arrays of size $1, \dots, k$
- Inductive step: For array A size $k + 1$
  - For any $k \geq 1$, we have $\frac{\lceil k+1 \rceil}{2} \leq k$
  - Inductive assumption implies that the two half arrays will be sorted correctly by merge sort,
  - since we know that the merge procedure will maintain the correct order, we see that merge_sort correctly sort A of size $k + 1$

Proof by induction: a general template

Theorem: $p(n)$ is true for every positive integer n

- Base case: **Prove** that $p(1)$ is true
  - This serves as the foundation for the inductive argument.
- Inductive step: **Prove** that $p(n)$ is true with the assumption that $p(k)$ is true for all $k < n$.

  Often broken down into two parts:

  - **Inductive assumption:** This step does not prove the statement for any n value, it merely states the assumption:

    $p(k)$ holds for any $k < n$

  - **Actual Proof:** show that

    $$p(n-1) \rightarrow p(n)$$ Simple induction

    Or

    $$p(1), \dots, p(n-1) \rightarrow p(n)$$ Strong induction

Simple example

$$1 + 2 + \cdots + n = \frac{n(n+1)}{2}$$

Base case:

Inductive step:

Example: Making Postage

- **Prove**: Any postage $\geq$ 20 cents can be made with 4 and 5 cents stamps.
- **Base case**: 20 cents can be made with 4 fives
- **Inductive hypothesis**: assume we can make postage 20,..,k
- **Inductive step**: show that we can make postage k+1
  - How can we reduce the problem to a smaller one such that it will be covered by the inductive hypothesis?
  - That is easy: we simply use one of the available stamps, e.g., a 4-cent stamp, we will then have a smaller problem of k+1-4=k-3
  - Since k-3 is less than k, thus covered by the inductive hypothesis, we know it is true. Hence, we can conclude that k+1 can be made with 4 and 5 cents stamps.

What went wrong in this proof?

Complete Proof

- **Base cases:**
  - **n=20**: 4 x5; **n=21**: 4x4+5; **n=22**: 3x4+2x5; **n=23**: 2x4+3x5
- **Inductive assumption:**

assume that any postage amount between $20 \ldots k$ can be made with 4, 5 cents stamps, where $k \geq 23$

- **Inductive step:**

We want to show that we can make k+1,

  - Use a 4-cent stamp, this reduces the posted to be made to $k - 3$
  - For $k \geq 23$, we have $20 \leq k - 3 < k$, which we know can be made due to the inductive assumption
  - Hence, we can make $k + 1$

QED

Another attempt at making postage

- Statement: Any postage $\geq$ 20 cents can be made with 5 and 6 cents stamps.
- What should the base cases be?

A. 20 cents

B. 20, 21, 22, 23 cents

C. 20, 21, 22, 23, 24 cents

D. 20, 21, 22, 23, 24, 25 cents

Inductive vs. Recursion

- Induction:

  prove $p(n)$

- If n is small, prove directly --- base case

- If n is large, prove with the assumption $p(k)$ is true for all $k < n$

- Recursion:

  solve problem of size n

- If n is small, solve directly --- base case

- If n is large, solve it using the solution of smaller problems

Recursive algorithm for postage

- Postage(n)

  if n = 20 return “5 4c”

  else if n= 21 return “4 4c, 1 5c”

  else if n = 22 return “3 4c, 2 5c”

  else if n= 23 return “4 4c, 3 5c”

  else return Postage(n-4) + 1 4c

If n is small, solve it directly

Otherwise, solving it problem using the solution to smaller problems

What is run time $T(n)$ for making postage n?

$$T(n) = T(n-4) + c$$

Solving Recurrence Relation with telescoping

$T(n) = T(n-4) + c$

$T(n-4) = T(n-8) + c$

$T(n) = T(n-8) + 2c$

$T(n-8) = T(n-12) + c$

……

$T(n) = T(n-4k) + kc$

How many layers of recursion ?

When do we hit the base case? $n - 4k \approx 23$

$$k_{max} \approx \frac{n-23}{4} = O(n)$$

$$T(n) = O(n)$$