The Substitution Method

- Induction requires us to show that the solution remains valid for the limit conditions

The Substitution Method

- Induction requires us to show that the solution remains valid for the limit conditions
- **Base case**: show that the inequality holds for some n sufficiently small

The Substitution Method

- Induction requires us to show that the solution remains valid for the limit conditions
- **Base case**: show that the inequality holds for some *n* sufficiently small
  - If $n = 1 \rightarrow T(1) \leq c * 1 * \log 1 = 0$

The Substitution Method

- Induction requires us to show that the solution remains valid for the limit conditions
- **Base case**: show that the inequality holds for some *n* sufficiently small
  - If $n = 1 \rightarrow T(1) \leq c * 1 * \log 1 = 0$
  - However, $T(n) = 2T(\lfloor n/2 \rfloor) + n \therefore T(1) = 1$

The Substitution Method

- Induction requires us to show that the solution remains valid for the limit conditions
- **Base case**: show that the inequality holds for some *n* sufficiently small
  - If $n = 1 \rightarrow T(1) \leq c * 1 * \log 1 = 0$
  - However, $T(n) = 2T(\lfloor n/2 \rfloor) + n \therefore T(1) = 1$
  - But

$n=2 \rightarrow T(2) =2T(1) + 2 = 4$ and $cn \log n = c * 2 * \log 2 = 2c$

$n=3 \rightarrow T(3)=2T(1) + 3 = 5$ and $cn \log n = c * 3 * \log 3$

The Substitution Method

- We can start from $T(2)=4$ or $T(3)=5$ using some $c \geq 2$, given that

$\boxed{T(n) \leq cn \log n}$ $\boxed{T(2) \leq c2 \log 2}$ $\boxed{T(3) \leq c3 \log 3}$

The Substitution Method

- We can start from $T(2)=4$ or $T(3)=5$ using some $c \geq 2$, given that

$T(n) \leq cn \log n$	$T(2) \leq c2 \log 2$	$T(3) \leq c3 \log 3$

- Base case holds w.r.t. the asymptotic notation:
  - $T(n) \leq cn \log n$ for $n \geq n_0$

The Substitution Method

- We can start from $T(2)=4$ or $T(3)=5$ using some $c \geq 2$, given that

$T(n) \leq cn \log n$	$T(2) \leq c2 \log 2$	$T(3) \leq c3 \log 3$

- Base case holds w.r.t. the asymptotic notation:
  - $T(n) \leq cn \log n$ *for* $n \geq n_0$
- Hint: extend the boundary conditions to make the inductive hypothesis count for small values of *n*

The Substitution Method

- **Inductive hypothesis:**
  - Assume that *T(n)* ≤ *c n log n* for *c > 0* holds for all positive *m* < *n*, in particular for *m* = $\lfloor n/2 \rfloor$, yielding

$$T(\lfloor n/2 \rfloor) \leq c \lfloor n/2 \rfloor \log(\lfloor n/2 \rfloor)$$

The Substitution Method

- Induction: Inequality holds for *n*

- $$\boxed{T(n) = 2T(\lfloor n/2 \rfloor) + n}$$ Original recurrence

$$\le 2(c\lfloor n/2 \rfloor \log\lfloor n/2 \rfloor) + n$$

$$\le cn \log(n/2) + n$$

$$= cn(\log n - \log 2) + n$$

$$= cn \log n - cn + n$$

$$\le cn \log n$$

The Substitution Method

- Induction: Inequality holds for *n*

$$T(n) = 2T(\lfloor n/2 \rfloor) + n$$

$$\leq 2\boxed{(c\lfloor n/2 \rfloor \log \lfloor n/2 \rfloor))} + n \quad \text{Inductive hypothesis}$$

$$\leq cn \log(n/2) + n$$

$$= cn(\log n - \log 2) + n$$

$$= cn \log n - cn + n$$

$$\leq cn \log n$$

The Substitution Method

- Induction: Inequality holds for *n*

$$T(n) = 2T(\lfloor n/2 \rfloor) + n$$

$$\leq 2(c\lfloor n/2 \rfloor \log \lfloor n/2 \rfloor)) + n$$

- $\leq cn \log(n/2) + n$ simplify expression

$$= cn(\log n - \log 2) + n$$

$$= cn \log n - cn + n$$

$$\leq cn \log n$$ ***(holds for c ≥1, upper bound analysis)***

The Substitution Method

- Induction: Inequality holds for *n*

$$T(n) = 2T(\lfloor n/2 \rfloor) + n$$

$$\leq 2(c\lfloor n/2 \rfloor \log \lfloor n/2 \rfloor)) + n$$

$$\leq cn \log(n/2) + n$$

- $= cn(\log n - \log 2) + n$ Logarithm property

$$= cn \log n - cn + n$$

$\leq cn \log n$ ***(holds for c ≥1, upper bound analysis)***

The Substitution Method

- Induction: Inequality holds for *n*

$$T(n) = 2T(\lfloor n/2 \rfloor) + n$$

$$\leq 2(c\lfloor n/2 \rfloor \log \lfloor n/2 \rfloor)) + n$$

$$\leq cn \log(n/2) + n$$

$$= cn(\log n - \log 2) + n$$

- $$= cn \log n - cn + n$$ log 2 = 1

$$\leq cn \log n$$ ***(holds for c ≥1, upper bound analysis)***

The Substitution Method

- Induction: Inequality holds for *n*

$$T(n) = 2T(\lfloor n/2 \rfloor) + n$$

$$\leq 2(c\lfloor n/2 \rfloor \log \lfloor n/2 \rfloor)) + n$$

$$\leq cn \log(n/2) + n$$

$$= cn(\log n - \log 2) + n$$

$$= cn \log n - cn + n$$

- $$\leq cn \log n$$ ***(holds for c ≥1, upper bound analysis)***

Making a good guess

- Guessing a solution takes **experience** / **creativity**
  - Use **heuristics** to help you become a good guesser
  - Use **recursion trees**

Making a good guess

- Guessing a solution takes **experience** / **creativity**
  - Use **heuristics** to help you become a good guesser
  - Use **recursion trees**
- Consider this example: $T(n) = 2T(\lfloor n/2 \rfloor + 17) + n$
  - What is your guess here? $T(n) = O(n \log n)$
  - The term “17” cannot substantially affect the solution

Making a good guess

- Guessing a solution takes **experience** / **creativity**
  - Use **heuristics** to help you become a good guesser
  - Use **recursion trees**
- Consider this example: $T(n) = 2T(\lfloor n/2 \rfloor + 17) + n$
  - What is your guess here? $T(n) = O(n \log n)$
  - The term “17” cannot substantially affect the solution
- Prove loose upper and lower bounds, e.g.:
  - Start with $T(n) = \Omega(n)$, then $T(n) = O(n^2)$ until we converge to $T(n) = O(n \log n)$

Changing Variables

- A little **algebraic manipulation** can make unknown recurrence similar to one you have seen before

Changing Variables

- A little **algebraic manipulation** can make unknown recurrence similar to one you have seen before

- $T(n) = 2T(\sqrt{n}) + \lg n$ *$m = \log n \;\therefore\; n = 2^m$*

$$T(2^m) = 2T(2^{m/2}) + m$$

$S(m) = 2S(m/2) + m$ similar to $T(n) = 2T(n/2) + n$

$$T(n) = T(2^m) = S(m)$$

$$= O(m \log m) = O(\log n \log \log n)$$

Changing Variables

- A little **algebraic manipulation** can make unknown recurrence similar to one you have seen before

$$T(n) = 2T(\sqrt{n}) + \lg n$$

m = log n .·. *n = 2*^{m}

- $$T(2^m) = 2T(2^{m/2}) + m$$

S(m)=T(2^{m}*)*

$$S(m) = 2S(m/2) + m$$ similar to $$T(n) = 2T(n/2) + n$$

$$T(n) = T(2^m) = S(m)$$

$$= O(m \log m) = O(\log n \log \log n)$$

Note that *log 2*^{m} = *m* and *log 2*$^{m/2}$ = *m/2*

Changing Variables

- A little **algebraic manipulation** can make unknown recurrence similar to one you have seen before

$$T(n) = 2T(\sqrt{n}) + \lg n$$ *m = log n .: n = 2*^{m}

$$T(2^m) = 2T(2^{m/2}) + m$$ *S(m)=T(2*^{m}*)*

- $S(m) = 2S(m/2) + m$ similar to $T(n) = 2T(n/2) + n$

$$T(n) = T(2^m) = S(m)$$

$$= O(m \log m) = O(\log n \log\log n)$$

Note that *log 2*^{m} *= m* and *log 2*$^{m/2}$ *= m/2*

Changing Variables

- A little **algebraic manipulation** can make unknown recurrence similar to one you have seen before

$$T(n) = 2T(\sqrt{n}) + \lg n$$

m = log n .:. n = 2^m

$$T(2^m) = 2T(2^{m/2}) + m$$

$S(m)=T(2^m)$

$$S(m) = 2S(m/2) + m \quad \text{similar to} \quad T(n) = 2T(n/2) + n$$

- $T(n) = T(2^m) = S(m)$

$$= O(m \log m) = O(\log n \log \log n)$$

Note that *log* $2^m = m$ and *log* $2^{m/2} = m/2$

Changing Variables

- A little **algebraic manipulation** can make unknown recurrence similar to one you have seen before

$T(n) = 2T(\sqrt{n}) + \lg n$ *m = log n* $\therefore$ *n =* 2^m

$T(2^m) = 2T(2^{m/2}) + m$ *S(m)=T(*2^m*)*

$S(m) = 2S(m/2) + m$ similar to $T(n) = 2T(n/2) + n$

$T(n) = T(2^m) = S(m)$

- $= O(m \log m) = O(\log n \log \log n)$

Note that *log* 2^m = *m* and *log* $2^{m/2}$ = *m/2*

Exercise: Recursive Binary Search

- Given the recurrence equation of the binary search:

$$T(n) = \begin{cases} 1 & \text{if } n = 1; \\ 1 + T(n/2) & \text{if } n > 1; \end{cases}$$

- Guess the solution is: *$T(n) = O(\log n)$*

- Prove that $T(n) \leq c \log n$ for some $c>0$ using induction and for all $n \geq n_0$

Exercise: Recursive Binary Search

- Step 1: Prove that the base case holds
  - 1a: If it does not hold for *n=0*, then change the boundary conditions
- Step 2: Prove that the inductive step holds
  - 2a: define the inductive hypothesis
  - 2b: substitute the inductive hypothesis in the original recurrence
  - 2c: manipulate the inequality to demonstrate that the inductive hypothesis holds for the given boundary conditions

Solution: Recursive Binary Search

- Step 1: Prove that the base case holds
  - *$n=2 \rightarrow T(2) = T(1) + 2 = 3$ and $c \log n = c * \log 2 = c$*
- Step 2: Prove that the inductive step holds
  - 2a: $T(n/2) \leq c \log n/2$
  - 2b: $T(n) = T(n/2) + 1 \rightarrow T(n) \leq c \log n/2 + 1$
  - 2c: $T(n) \leq c \log n - c \log 2 + 1$
  - 2c: $T(n) \leq c \log n - c + 1$
  - 2c: $T(n) \leq c \log n$ (holds for $c \geq 1$)

Summary

- Many useful algorithms are recursive in structure:
    - to solve a given problem, they call themselves recursively one or more times to deal with closely related sub-problems
- We also analysed recurrences using the substitution method
- We have demonstrated that $T(n)$ of merge sort is $\Theta(n \log n)$, where $\log n$ stands for $\log_2 n$