- 教材答疑和讨论
  - TC第1、2、3章

# 问题1：计算问题与算法

- 你如何理解a well-specified computational problem？
  - Input + output + their relationship
- 你能举个例子吗？

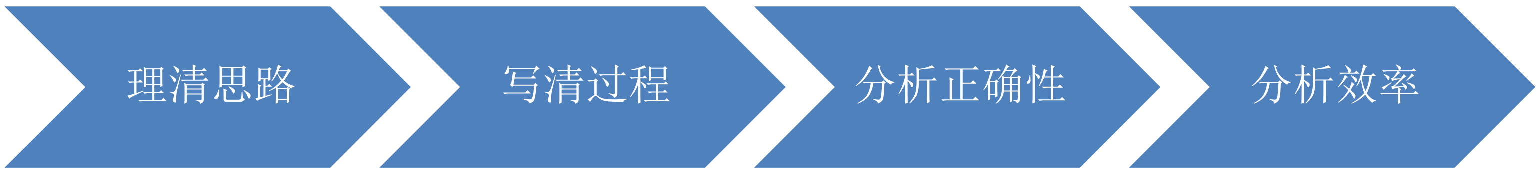
- 你如何理解an algorithm？
  - Well-defined computational procedure for achieving an input-output relationship
- 你能举个例子吗？

# 问题2：好算法

- 回忆一下，你写过哪些糟糕的算法？

- 一个好算法应具有哪些要素？
  - 正确的：总能停止且结果正确
  - 高效的：运行速度快、占用空间少
  - 易实现的：描述清晰、简单

- 如何设计出一个好算法？

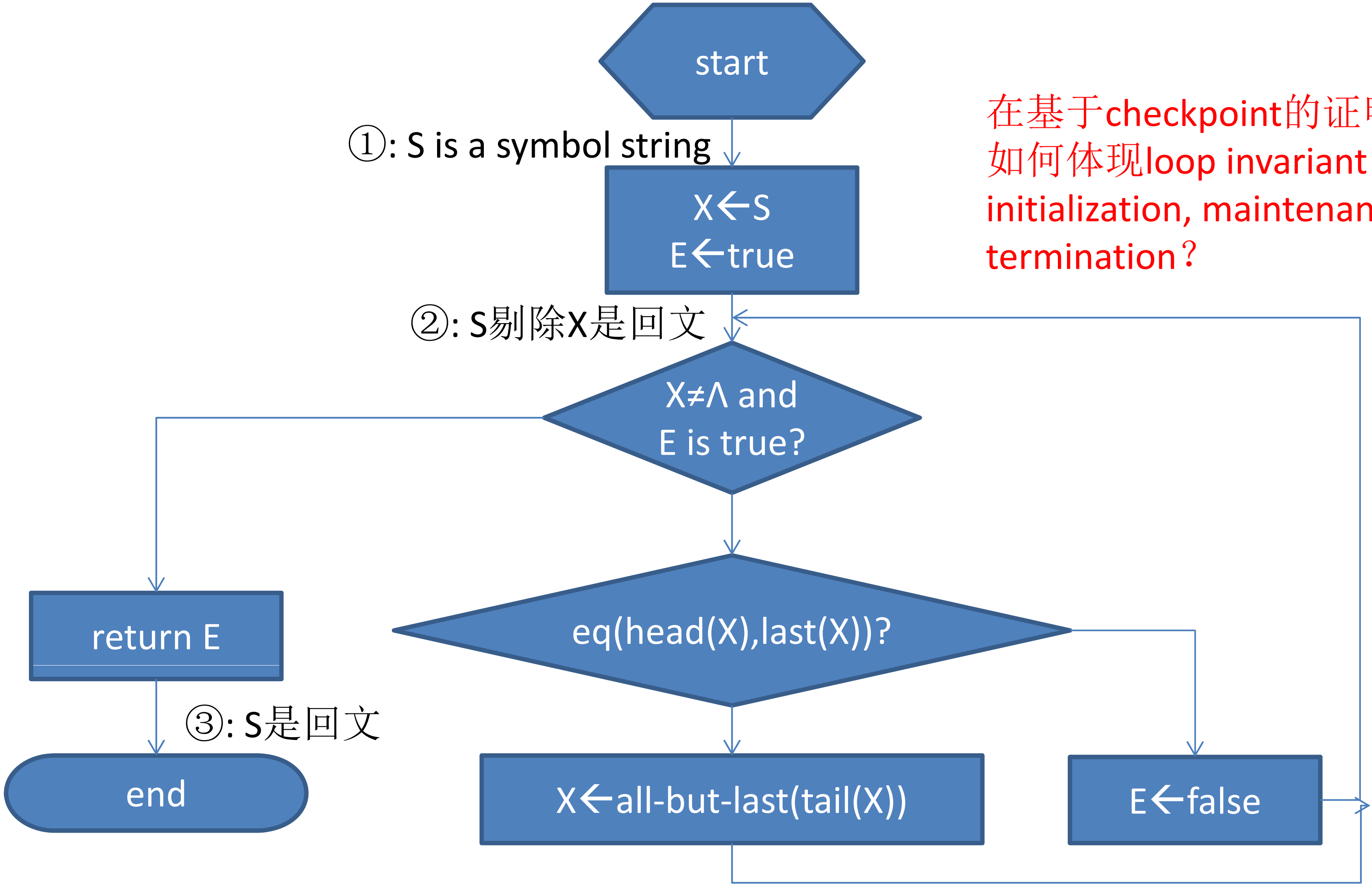


# 问题3：算法的正确性分析

- 如何证明算法是partially correct？
  1. 设置checkpoint
     - start后和end前各一个
     - 每个回路上至少一个（通常是第一次进入回路时）
  2. 为每个checkpoint设置invariant（最后一个invariant是算法期望的结果）
  3. 检查所有checkpoint之间的路径，说明为什么路径起点的invariant成立时，路径终点的invariant也成立

- 如何证明算法是totally correct？
  - Partially correct + termination

# 举例：回文检测算法




在基于checkpoint的证明中，如何体现loop invariant的initialization, maintenance和termination？

# 问题4：算法的效率分析

- 分析算法的效率时，为什么要先定义计算模型？

- Random-Access Machine有哪些要素？
  - 数据
    - 支持的类型
    - 存储的方式
  - 指令
    - 支持的类型
    - 执行的方式

# 举例：回文检测算法

| PALINDROME-TEST (S) | cost | times |
|---|---|---|
| 1. X=S | $c_1$ | 1 |
| 2. E=true | $c_2$ | 1 |
| 3. while X≠Λ and E=true | $c_3$ | p |
| 4. if eq(head(X),last(X))=true | $c_4$ | p-1 |
| 5. X=all-but-last(tail(X)) | $c_5$ | q |
| 6. else | $c_6$ | p-1-q |
| 7. E=false | $c_7$ | p-1-q |
| 8. return E | $c_8$ | 1 |

input size是什么？
如何计算running time？

$$running\ time = c_1 + c_2 + c_3 \cdot p + c_4 \cdot (p-1) + c_5 \cdot q + c_6 \cdot (p-1-q) + c_7 \cdot (p-1-q) + c_8$$

best case: $c_1 + c_2 + c_3 \cdot 2 + c_4 \cdot 1 + c_5 \cdot 0 + c_6 \cdot 1 + c_7 \cdot 1 + c_8$

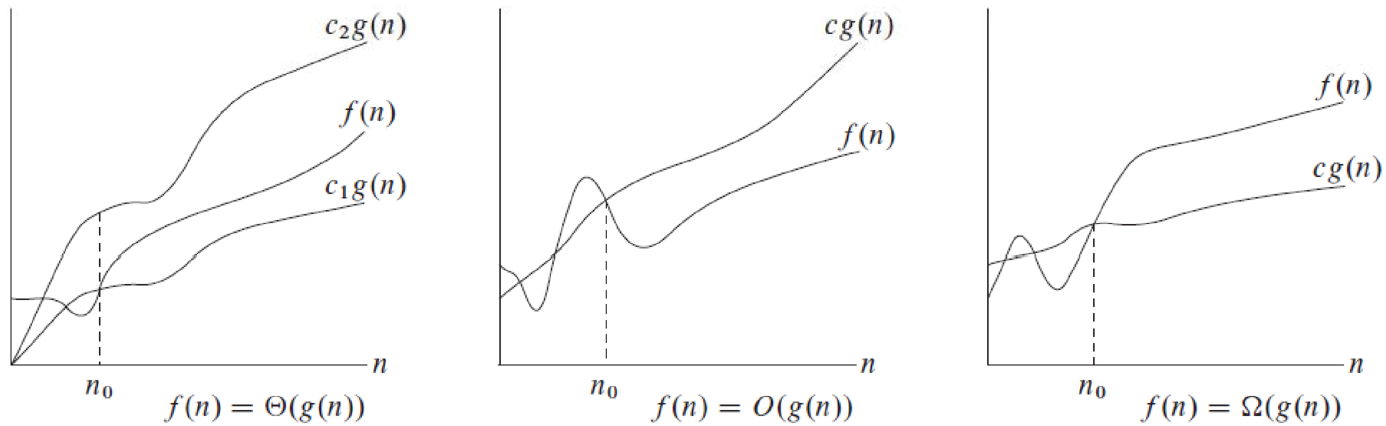
worst case: $c_1 + c_2 + c_3 \cdot \left(\left\lceil \frac{n}{2} \right\rceil + 1\right) + c_4 \cdot \left\lceil \frac{n}{2} \right\rceil + c_5 \cdot \left\lceil \frac{n}{2} \right\rceil + c_6 \cdot 0 + c_7 \cdot 0 + c_8$

average case?

- 为什么我们最关注worst case？
  - Gives us an upper bound on the running time for any input.
  - Occurs fairly often.
  - The “average case” is often roughly as bad as the worst case.

# 问题5：算法效率的渐进表示法

- Θ, O和Ω的本质是什么？
  - 函数的集合
- 你能具体解释Θ, O和Ω的含义吗？




$\Theta(g(n)) = \{f(n) :$ there exist positive constants $c_1$, $c_2$, and $n_0$ such that $0 \le c_1 g(n) \le f(n) \le c_2 g(n)$ for all $n \ge n_0\}$ .[1]

$O(g(n)) = \{f(n) :$ there exist positive constants $c$ and $n_0$ such that $0 \le f(n) \le cg(n)$ for all $n \ge n_0\}$ .

$\Omega(g(n)) = \{f(n) :$ there exist positive constants $c$ and $n_0$ such that $0 \le cg(n) \le f(n)$ for all $n \ge n_0\}$ .

⇨

$$\lim_{n\to\infty}\frac{f(n)}{g(n)} = c \in (0,\infty)$$

$$\lim_{n\to\infty}\frac{f(n)}{g(n)} < \infty$$

$$\lim_{n\to\infty}\frac{f(n)}{g(n)} > 0$$

# 举例：回文检测算法

best case: $c_1 + c_2 + c_3 \cdot 2 + c_4 \cdot 1 + c_5 \cdot 0 + c_6 \cdot 1 + c_7 \cdot 1 + c_8 \in \Theta(1)$

worst case: $c_1 + c_2 + c_3 \cdot \left( \left\lceil \frac{n}{2} \right\rceil + 1 \right) + c_4 \cdot \left\lceil \frac{n}{2} \right\rceil + c_5 \cdot \left\lceil \frac{n}{2} \right\rceil + c_6 \cdot 0 + c_7 \cdot 0 + c_8 \in \Theta(n)$

- 为什么低阶项和系数都可以忽略？
  - $c_1 n^2 \le an^2 + bn + c \le c_2 n^2$
- 如果改用O，有什么好处？
- 如何理解$an^2+bn+c=an^2+\Theta(n)$？这样改写有什么好处？

- O和o的区别是什么？

$$O(g(n)) = \{f(n) : \text{there exist positive constants } c \text{ and } n_0 \text{ such that } 0 \le f(n) \le cg(n) \text{ for all } n \ge n_0\}.$$

$$o(g(n)) = \{f(n) : \text{for any positive constant } c > 0, \text{ there exists a constant } n_0 > 0 \text{ such that } 0 \le f(n) < cg(n) \text{ for all } n \ge n_0\}.$$

$\Rightarrow$

$$\lim_{n\to\infty}\frac{f(n)}{g(n)} < \infty$$

$$\lim_{n\to\infty}\frac{f(n)}{g(n)} = 0$$

- 一个比喻

$$f(n) = O(g(n)) \quad \text{is like} \quad a \le b\,,$$
$$f(n) = \Omega(g(n)) \quad \text{is like} \quad a \ge b\,,$$
$$f(n) = \Theta(g(n)) \quad \text{is like} \quad a = b\,,$$
$$f(n) = o(g(n)) \quad \text{is like} \quad a < b\,,$$
$$f(n) = \omega(g(n)) \quad \text{is like} \quad a > b\,.$$

- 所以，具有哪些类似的性质？
  - 自反性
  - 对称性
  - 传递性
  - ……
- 但是，以下性质不成立，你能举个反例吗？

**Trichotomy:** For any two real numbers $a$ and $b$, exactly one of the following must hold: $a < b$, $a = b$, or $a > b$.

- 它们之间的关系是什么？