ثابت حلقه: نوعی اثبات استقرایی است

Loop-Invariant

- حقیقتی ثابت در آغاز حلقه درست است.
- حقیقتی ثابت در هر تکرار حلقه درست است.
- حقیقتی ثابت در انتهای حلقه درست است.

یادآوری روی اثبات استقرایی: فرض کنید $P(n)$ حکمی در مورد اعداد طبیعی باشد. برای اثبات صحت حکم برای تمام مقادیر طبیعی n، یک اثبات استقرایی صحت گزاره‌های زیر را تایید می‌کند:

* پایه استقرا: $P(1)$ درست است.

* گذار (گام) استقرا: به ازای عدد طبیعی مانند k، اگر $P(k)$ درست باشد آنگاه $P(k+1)$ نیز درست است.

در نهایت با تایید مراحل پایه و گام استقرا، صحت حکم به استقرا ثابت می‌شود.

گام‌های مورد نیاز برای اثبات درستی الگوریتم به روش ثابت حلقه:

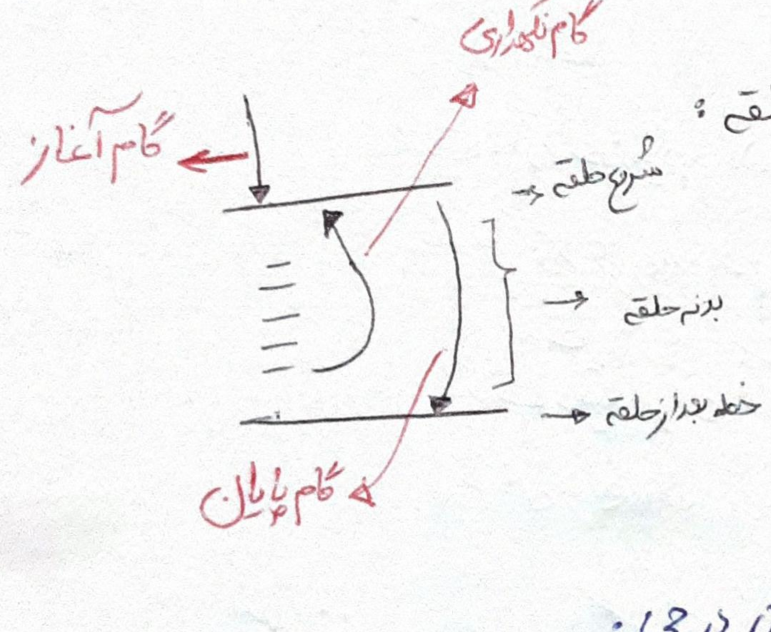
* گام یا آغاز (initialization): همانند بررسی پایه برای برای اثبات استقرایی است. در واقع باید نشان دهیم که حقیقت مورد نظر قبل از اجرای حلقه برقرار است.

* گام ۲، نگهداری (Maintenance): همانند بررسی گام گذار برای اثبات استقرایی است.

در واقع باید نشان دهیم که حقیقه ناوردای مورد نظر قبل از اجرای هر تکرار برقرار است.

به عبارت دیگر، قبل از تکرار حلقه درست است و قبل از تکرار حلقه بعد هم درست است.

* گام ۳، پایان (Termination): در پایان نشان می‌دهیم که پس از خارج شدن الگوریتم از حلقه، حقیقه ناوردایی دارای ویژگی است که می‌توان از آن صحت الگوریتم را استنتاج کرد.



جمع‌بندی نقش روی گام‌های ثابت حلقه:

① نکته مهم: تعیین حقیقه ثابت
گام‌های اثبات

② طی کردن گام‌های ثابت حلقه

اثبات درستی الگوریتم مرتب‌سازی درجی:

۱. تعیین حقیقه ناوردایی حلقه: درست پس از مقداردهی متغیر حلقه for (یعنی k) k-1 عنصر اول آرایه مرتب شده هستند. به عبارت دیگر A[1..k-1] مرتب است.

۲) طی کردن گام های ناوردایی حلقه:

گام آغاز: باید نشان دهیم که حقیقت ناوردایی حلقه، درست قبل از اولین مقدار دهی

به K برقرار است ⟵ در این مرحله $K=2$ است و $A[1..K-1] = A[1..1]$

چون ۱ عنصر دارد مرتب است.

گام نگهداری: باید نشان دهیم که در آغاز هر تکرار، حقیقت ناوردایی برقرار است.

به عبارت دیگر اگر تکرار i ام انجام شده باشد و در آغاز تکرار i+1 ام

باشیم باید نشان دهیم که زیرآرایه $\underbrace{A[1..K]}_{\text{یعنی } K=i+1} = A[1..i]$ مرتب

است. از آنجا که حلقه های ۲ تا ۷ مقدار $A[i]$ را در جای

صحیح خود قرار می دهد پس $A[1..i]$ موقت مرتب است.

نکته: اثبات دقیق تر آن است که حقیقت ناوردایی نیز برای حلقه

داخلی (while) نیز برقرار شود.

گام پایان: در پایان بررسی می کنیم که وقتی حلقه به پایان رسید، چه روی می دهد.

شرط پایان حلقه ⟵ $\underline{K=n+1}$ ⟵ $A[1..n]$ به صورت

مرتب شده می باشد

⟵ به وضوح حقیقت ناوردایی حلقه، صحت الگوریتم را

نتیجه می دهد.

① تحلیل پیچیدگی زمانی الگوریتم مرتب سازی درجی :

c_i هزینه اجرای بار خط i

```
Insertion-Sort ( A[1..n])

1.  for k=2 to n do              c1
2.      key ← A[k]               c2
3.      i ← k                    c3
4.      while i>1 and A[i-1]> key do   c4
5.          A[i] ← A[i-1]        c5
6.          i ← i-1              c6
7.      A[i] ← key               c7
```

$$T(n) = \underbrace{(n-2+1+1)\,c_1}_{\text{line 1}} + \underbrace{(n-1)(c_2+c_3)}_{\text{lines 2,3}} + \underbrace{\left(\sum_{k=2}^{n} t_k\right) c_4}_{\text{line 4}} + \underbrace{\left(\sum_{k=2}^{n} (t_k-1)\right)(c_5+c6)}_{\text{lines 5,6}} + \underbrace{(n-1)\,c7}_{\text{line 7}} \to A^*$$

تابع کامل پیچیدگی زمانی

یک فرم کلی $\to T(n) = An + B + \left\{ c \sum_{k=2}^{n} t_k \right\} \to$ چگونه محاسبه کنیم؟

اینجاست که تکنیک سه حالت برای تحلیل کار ایجاد داریم

- بهترین حالت
- بدترین حالت
- حالت میانگین

(۴) * بهترین حالت ← مرتب شده صعودی باشد ← فقط خط ۴ اجرا می‌شود و داریم

$$T(n) = A^* + \underbrace{(n-1)C4}_{\text{line } 4} + \underbrace{0}_{\text{line } 5,6} \qquad (t_k = 1)$$

فرم کلی: $T(n) = An + B$

(۳) * بدترین حالت ← مرتب شده نزولی باشد ←

~~خط 6,5 → $k=2 \to 1$, $k=3 \to 2$, ⋮, $k=n \to n-1$~~ ~~$\sum_{k=2}^{n} t_k = \frac{n(n-1)}{2}$~~

خط 4 → $\left.\begin{array}{l} k=2 \to 2 \\ k=3 \to 3 \\ \vdots \\ k=n \to n \end{array}\right\} \quad \sum_{k=2}^{n} t_k = \frac{n(n+1)}{2} - 1$

خط 6,5 → $\sum_{k=2}^{n} (t_k - 1) = \frac{(n-1)n}{2}$

$$\downarrow$$

$$\left\{\begin{array}{l} 1 \\ 2 \\ \vdots \\ n-1 \end{array}\right.$$

$$T(n) = A^* + \left(\frac{n(n+1)}{2} - 1\right)C4 + \frac{(n-1)n}{2}(C5 + C6)$$

فرم کلی: $T(n) = An + B + Cn^2$