



پاسخ تمرین ها سری سوم

مبانی برنامه نویسی (دکتر ابریشمی) - مهر ۱۴۰۰

مطالب

01

$\text{fof} \dots \text{fof}(x)$

03

مبنای آینه‌ای

02

اعداد دست یافتنی

04

متادوز

مزایای استفاده از زیرالگوریتم ها

جلوگیری از تکرار اضافی الگوریتم ها

ساده شدن عیب یابی و اشکال زدایی الگوریتم

بالا رفتن خوانایی برنامه

امکان تقسیم کار به چند بخش و واگذاری آن به افراد مختلف





01

`fof...fof(x)`



برای نوشتن الگوریتم ابتدا خود، مسئله را حل میکنیم.
مسئله را به چند زیر مسئله تقسیم میکنیم.

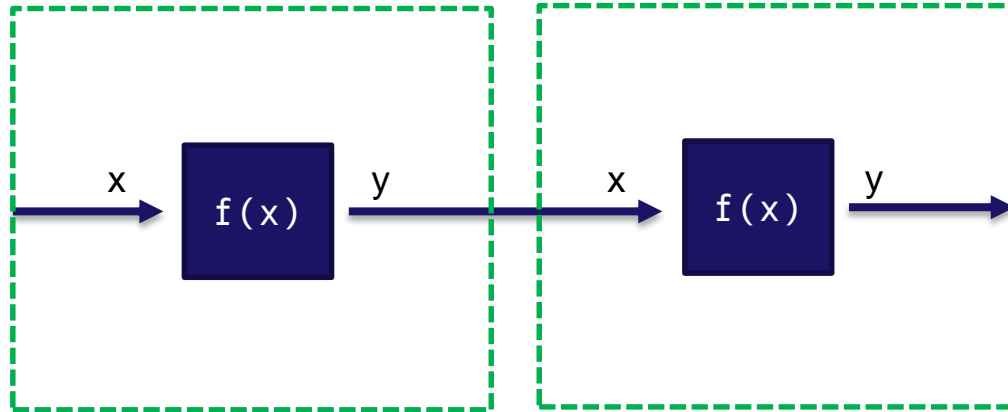
$f(x)$



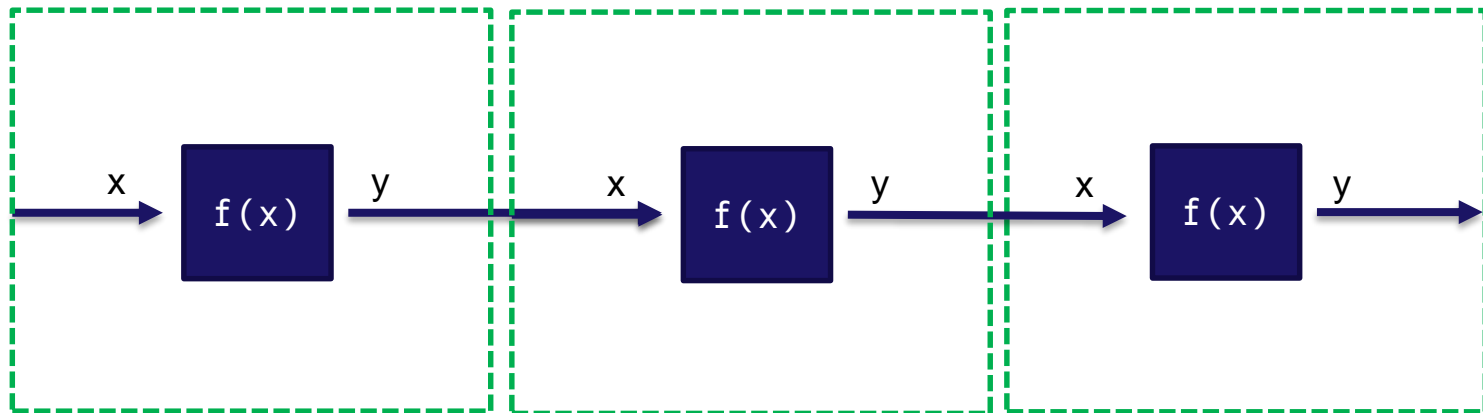
$f \circ f(x)$



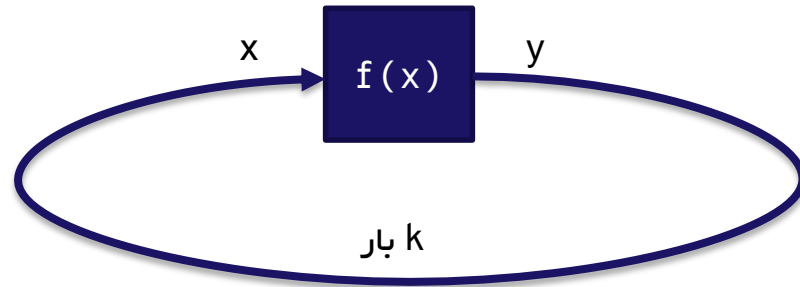
$\text{fof}(x)$



fofof(x)



$f_k f(x, k)$



زیر الگوریتم $f(x)$

$$f \leftarrow x * x + 2 * x + 2 \quad .1$$

2. برگشت

زیرالگوریتم $f^k f(x, k)$

1. $i \leftarrow 1, y \leftarrow x$

2. تا زمانی که $(i \leq k)$ دستورات ۳ تا ۴ را تکرار کن:

3. $y \leftarrow f(y)$

4. $i \leftarrow i + 1$

5. $f^k f \leftarrow y$

6. برگشت

الگوریتم اصلی

1. x و k را بخوان.

2. $result \leftarrow fkf(x, k)$

3. $result$ را چاپ کن.

4. پایان.



02

اعداد دست یافتنی

زیر الگوریتم $f(x)$

1. $f \leftarrow \text{removeRightZeros}(x + 1)$

2. برگشت

زیرالگوریتم
`removeRightZeros(x)`

شرایط استثناء را در نظر
بگیرید.

1. $y \leftarrow x$
2. تا زمانی که $(y \% 10 = 0)$ و $(y \neq 0)$ دستور ۳ را تکرار کن:

3. $y \leftarrow y/10$

4. $removeRightZeros \leftarrow y$

5. برگشت

زیرالگوریتم $f(x)$

1. $f \leftarrow \text{removeRightZeros}(x + 1)$

2. برگشت

زیر الگوریتم
 $isReproduced(x, A)$

1. $i \leftarrow 1$
2. تا زمانی که $(i \leq 1000)$ دستورات ۳ تا ۴ را تکرار کن:
3. اگر $x = A(i)$ آنگاه $isReproduced \leftarrow 1$ و برگشت
4. $i \leftarrow i + 1$
5. $isReproduced \leftarrow 0$
6. برگشت

الگوریتم اصلی

1. x را بخوان.
2. آرایه A را ۱۰۰۰ عضو در نظر بگیرید.
3. $count \leftarrow 0, repeated \leftarrow 0$
4. تا زمانی که $(repeated \neq 1)$ دستورات ۵ تا ۷ را تکرار کن:
5. $y \leftarrow f(y)$
6. $repeated \leftarrow isReproducted(y, A)$
7. اگر $repeated = 0$ آنگاه :
 - $count \leftarrow count + 1$
 - $A(count) \leftarrow y$
8. پایان



03

مبنای آینه ای

زیر الگوریتم
`bBaseToDecimal(a,b)`

1. $result \leftarrow 0, pow \leftarrow 1$

2. تا زمانی که $(a \neq 0)$ دستورات ۳ تا ۵ را تکرار کن:

3. $result \leftarrow result + (a \% 10) * pow$

4. $a \leftarrow a / 10$

5. $pow \leftarrow pow * b$

6. $bBaseToDecimal \leftarrow result$

7. برگشت

زیر الگوریتم
`decimalToCBase(c,d)`

1. $result \leftarrow 0, pow \leftarrow 1$

2. تا زمانی که $(d \neq 0)$ دستورات ۳ تا ۵ را تکرار کن:

3. $result \leftarrow result + (d \% c) * pow$

4. $d \leftarrow d / c$

5. $pow \leftarrow pow * 10$

6. $DecimalToCBase \leftarrow result$

7. برگشت

زیر الگوریتم
`isPalindrome(x)`

1. $result \leftarrow 0, y \leftarrow x$
2. تا زمانی که $(y > 0)$ دستورات ۳ تا ۴ را تکرار کن:
3. $result \leftarrow result * 10 + (a \% 10)$
4. $y \leftarrow y / 10$
5. اگر $result = x$ آنگاه $isPalindrome \leftarrow 1$
- در غیر این صورت $isPalindrome \leftarrow 0$
6. برگشت

الگوریتم اصلی

1. a، b، c را بخوان.

2. $d \leftarrow bBaseToDecimal(a,b)$

3. $x \leftarrow decimalToCBase(c,d)$

4. اگر $isPalindrome(x) = 1$ آنگاه چاپ کن "YES"

در غیر اینصورت چاپ کن "NO"

5. پایان.

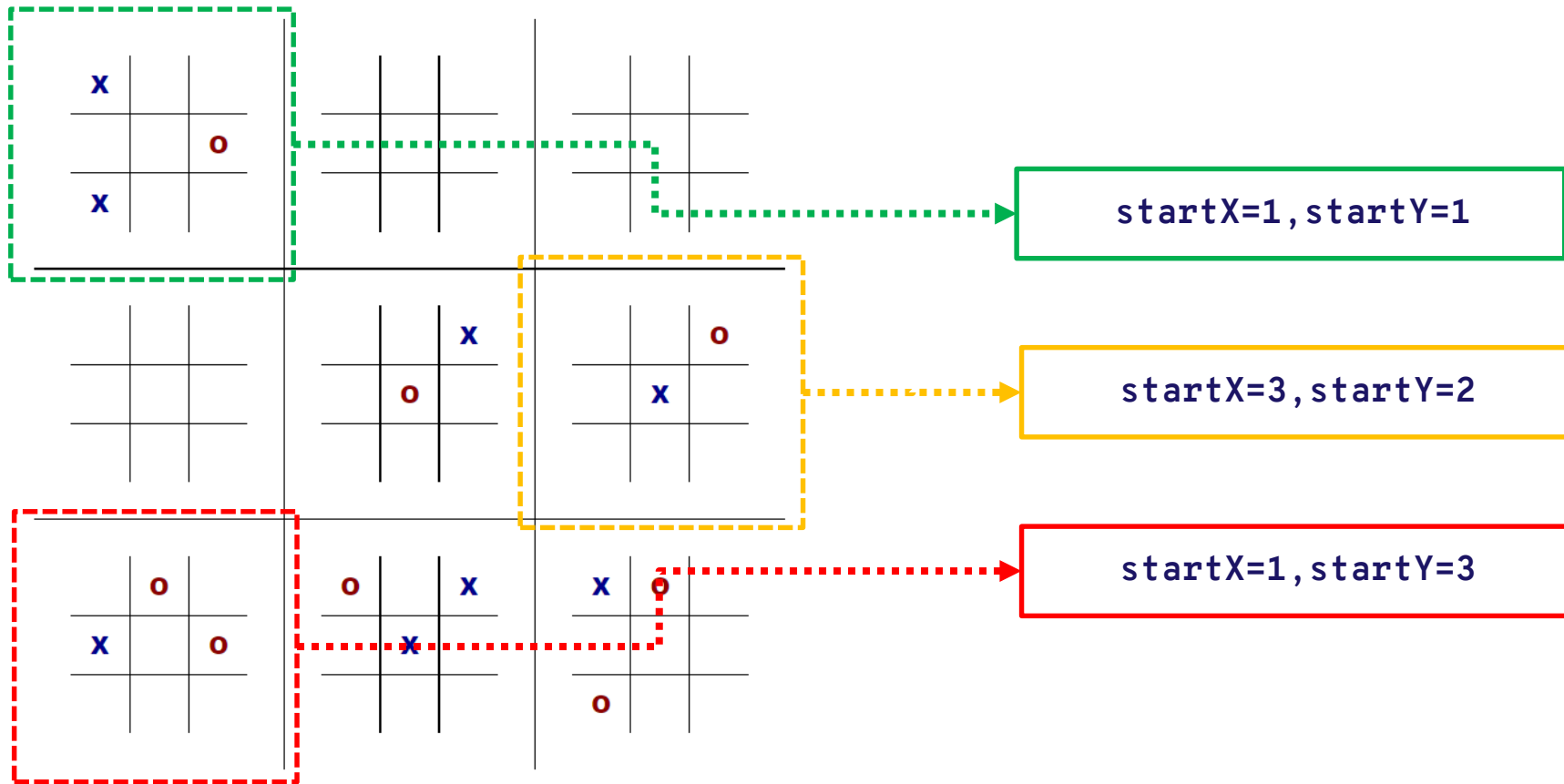


04

متادوز



زیر الگوریتم getWinner مختصات شروع یک دوز ۳*۳ و یک آرایه که شامل این دوز است را بعنوان پارامتر ورودی دریافت میکند. سپس با چک کردن ستون ها، ردیف ها و قطر ها برنده را مشخص می کند و مقدار آن را باز می گرداند. اگر برنده ای وجود نداشته باشد و تمام خانه های آن دوز پر شده باشد مقدار صفر را به عنوان مساوی باز می گرداند.



زیر الگوریتم

`getWinner(startX, startY, board)`

1. $winner \leftarrow -1$
2. $winner \leftarrow checkCols(startX, startY, board)$
3. اگر $winner \neq -1$ آنگاه $getWinner \leftarrow winner$ ، برگشت.
4. $winner \leftarrow checkRows(startX, startY, board)$
5. اگر $winner \neq -1$ آنگاه $getWinner \leftarrow winner$ ، برگشت.
6. $winner \leftarrow checkDim(startX, startY, board)$
7. اگر $winner \neq -1$ آنگاه $getWinner \leftarrow winner$ ، برگشت.
8. اگر $isFull(startX, startY, board) = 1$ آنگاه $getWinner \leftarrow 0$ ، برگشت.
9. $getWinner \leftarrow -1$
10. برگشت

زیرالگوریتم `checkCols(startX, startY, board)`

1. $i \leftarrow 0$

2. تا زمانی که $i < 3$ دستور ۳ تکرار کن:

3. اگر $board(startX + i, startY) = board(startX + i, startY + 1)$ و

$board(startX + i, startY + 2) = board(startX + i, startY + 1)$ و

$board(startX + i, startY) \neq 0$ آنگاه

$checkCols \leftarrow board(startX + i, startY)$ ، برگشت.

4. $checkCols \leftarrow -1$

5. برگشت

اگر برنده ای وجود نداشته باشد
مقدار -۱ را باز می گردانند.

زیر الگوریتم
`checkRows (startX, startY, board)`

1. $i \leftarrow 0$

2. تا زمانی که $i < 3$ دستور ۳ تکرار کن:

3. اگر $board(startX, startY + i) = board(startX + 1, startY + i)$ و

$board(startX + 2, startY + i) = board(startX + 1, startY + i)$ و

$board(startX, startY + i) \neq 0$ آنگاه

$checkRows \leftarrow board(startX, startY + i)$ ، برگشت.

4. $checkRows \leftarrow -1$

5. برگشت

زیر الگوریتم

`checkDim(startX, startY, board)`

1. اگر $board(startX, startY) = board(startX + 1, startY + 1)$ و

و $board(startX + 2, startY + 2) = board(startX, startY)$

آنگاه $board(startX, startY) \neq 0$

$checkDim \leftarrow board(startX, startY)$ ، برگشت.

2. اگر $board(startX + 2, startY) = board(startX + 1, startY + 1)$ و

و $board(startX, startY + 2) = board(startX + 1, startY + 1)$

آنگاه $board(startX + 1, startY + 1) \neq 0$

$checkDim \leftarrow board(startX + 1, startY + 1)$ ، برگشت.

3. $checkDim \leftarrow -1$

4. برگشت

زیرالگوریتم
`isFull(startX, startY, board)`

1. $i \leftarrow 0$

2. تا زمانی که $i < 3$ دستورات ۳ تا ۵ تکرار کن:

3. $j \leftarrow 0$

4. تا زمانی که $j < 3$ دستور ۵ تکرار کن:

5. اگر $board(startX + i, startY + j) = 0$ آنگاه

$isFull \leftarrow 0$ ، برگشت.

6. $isFull \leftarrow 1$

7. برگشت



برای قرار دادن علامت هر بازیکن در آرایه یک عدد را
به هر بازیکن نظیر می‌کنیم:

- `XPlayer` \equiv 1
- `OPlayer` \equiv 2

✓ برای خانه‌های خالی عدد صفر را در نظر می‌گیریم.

الگوریتم اصلی

1. شروع
2. $turn \leftarrow 0$
3. $gameWinner \leftarrow -1$
4. $OPlayer \leftarrow 2, XPlayer \leftarrow 1$
5. آرایه $board$ را با ابعاد $9 * 9$ در نظر بگیر و تمام عناصر آن را برابر ۰ قرار بده
6. آرایه $mainBoard$ را با ابعاد $3 * 3$ در نظر بگیر و تمام عناصر آن را برابر ۰ قرار بده
7. تا زمانی که $gameWinner = -1$ دستورات ۸ تا ۱۵ را تکرار کن:
8. x را بخوان
9. y را بخوان
10. اگر $turn \% 2 = 0$ آنگاه $board(x,y) \leftarrow XPlayer$ در غیر اینصورت $board(x,y) \leftarrow OPlayer$
11. $boardx \leftarrow (x - 1) / 3 + 1$
12. $boardy \leftarrow (y - 1) / 3 + 1$

ادامه ...

ادامه الگوریتم اصلی

13. $winner \leftarrow getWinner(boardx, boardy, board)$

14. اگر $winner \neq -1$ آنگاه $mainBoard(boardx, boardy) \leftarrow winner$

15. $gameWinner \leftarrow getWinner(mainBoard, 1, 1)$

16. اگر $gameWinner = XPlayer$ آنگاه چاپ کن X won!

17. اگر $gameWinner = OPlayer$ آنگاه چاپ کن O won! در غیر این صورت چاپ کن $Draw!$

18. پایان.