

هو العليم



الگوریتم و محاسبه

نیمسال دوم سال تحصیلی 1403 – 1402

تمرینات 1

1. این الگوریتم را، که با راهبردی ساده‌اندیشانه مسأله‌ای را حل می‌کند، در نظر بگیرید.

ALGORITHM *Example*(a, b)

// Input: Two integers a and $b > 0$

$x \leftarrow 0$

$y \leftarrow |a|$

while $y \geq b$ **do**

```

y ← y - b
x ← x + 1
if a < 0 and y = 0
  x ← -x
if a < 0 and y > 0
  y ← b - y
  x ← -(x + 1)
return (x, y)

```

الف) با یک مثال عددی، مسأله‌ای را که الگوریتم حل می‌کند، مشخص کنید.

ب) درستی الگوریتم را ثابت کنید.

پ) با این فرض که $a > b$ باشد، درستی این ادعا را که کارایی زمانی الگوریتم $O(x \log a)$ است، توجیه کنید.

جواب:

مراجع:

2. درستی هر یک از این ادعاهای ریاضی را ثابت کنید.

الف)

$$\sum_{i=0}^n a^i \in \Theta(a^n), \quad a > 1$$

ب)

$$\sum_{i=1}^n i^k \in \Theta(n^{k+1})$$

پ)

$$\sum_{i=1}^n \frac{1}{i} \in \Theta(\ln n)$$

(ت)

$$\sum_{i=0}^n \binom{n}{i} i \in \Theta(n2^n)$$

جواب:

مراجع:

3. این سه عملیات را در نظر بگیرید:

- Push(S, e) : عنصری را در انتهای یک ساختار داده درج می‌کند.
 - Pop(S) : عنصری را از انتهای یک ساختار داده حذف می‌کند.
 - Find – Min(S) : کوچک‌ترین عنصر را در یک ساختار داده برمی‌گرداند (بدون آنکه آن را حذف کند).
- در اینجا ما به دنبال طراحی الگوریتم‌هایی برای انجام سریع سه عملیات Push و Pop و Find – Min هستیم. و قسمت عمده طراحی این الگوریتم‌ها، چیزی نیست جز طراحی ساختار داده‌ای برای چینش مناسب داده‌ها. ساختار داده‌ای طراحی کنید که با آن بتوان، هر سه عملیات Push و Pop و Find – Min را در زمان $\Theta(1)$ انجام داد. بعد از طراحی ساختار داده، شبیه‌کد الگوریتم‌های Push(S, e) و Pop(S) و Find – Min(S) را بنویسید.

جواب:

مراجع:

4. فرض کنید A مجموعه‌ای متناهی باشد، مثلاً مجموعه $A = \{1, 2, \dots, n\}$ ؛ و f تابعی باشد از A به A :

$$f: A \rightarrow A$$

الف) الگوریتمی کارا طراحی کنید که با آن بتوان تعیین کرد که آیا چنین توابع f ای، «یک - به - یک» هستند یا خیر. الگوریتمتان را با شبکه‌کد توصیف کنید و کارایی زمانی آن را نیز اندازه بگیرید.

ب) الگوریتمی کارا طراحی کنید که با آن بتوان بزرگترین زیرمجموعه $S \subseteq A$ را به گونه‌ای که تابع

$$f: S \rightarrow S$$

«یک - به - یک» باشد، تعیین کرد. الگوریتمتان را با شبکه‌کد توصیف کنید و کارایی زمانی آن را نیز اندازه بگیرید.

جواب:

مراجع:

5. این الگوریتم بازگشتی برای مسأله یکتایی عناصر را در نظر بگیرید.

ALGORITHM *UniqueElements*($A[0..n - 1]$)

// Determines whether all the elements in a given array are distinct

// Input: An array $A[0..n - 1]$

// Output: Returns “true” if all the elements in A are distinct and “false” otherwise

if $n = 1$

return true

else if not *UniqueElements*($A[0..n - 2]$)

return false

else if not *UniqueElements*($A[1..n - 1]$)

return false

else

return $A[0] \neq A[n - 1]$

الف) چرا این الگوریتم بازگشتی، جواب درست مسأله را برمی‌گرداند؟

ب) کارایی زمانی الگوریتم چقدر است؟ چرا الگوریتم ناکارا است؟

پ) یک الگوریتم بازگشتی کارا برای مسأله طراحی کنید و کارایی زمانی آن را نیز اندازه بگیرید.

جواب:

مراجع:

