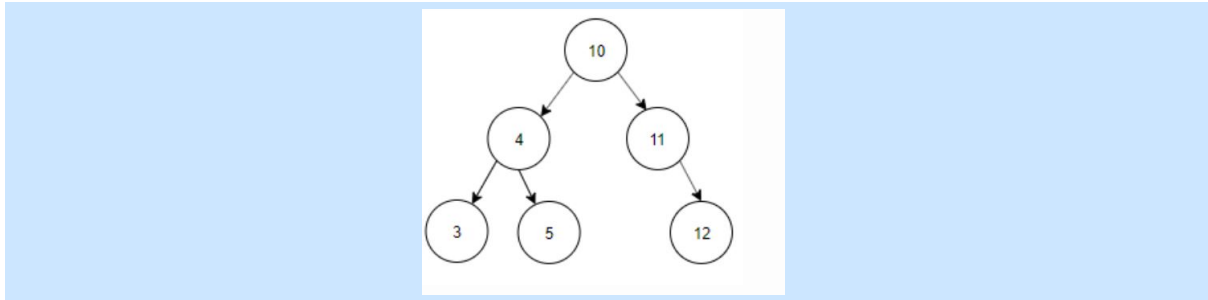


هر سوال را در محل در نظر گرفته شده پاسخ دهید. پاسخ های خارج از محل تصحیح نمیشوند.
غزاله محمودی - سید صالح اعتمادی

۱. [۱۰] پیمایش *preorder* یک Binary Search Tree به صورت زیر است. درخت متناظر با آن را رسم کنید.

(10, 4, 3, 5, 11, 2)



۲. [۲۰] به سوالات زیر پاسخ کوتاه دهید.

(a) What is the size of the array needed to store international phone numbers with up to 15 digits using direct addressing

10^{15}

(b) What is the average running time of Rabin-Karp's algorithm for text T and pattern P if there are no occurrences of the pattern in the text?

$O(|P| + |T|)$

۳. [۲۰] پیچیدگی محاسباتی عملیات های زیر را در بدترین حالت و حالت متوسط/سرشکن همراه دلیل آنها بنویسید.

(a) Find in Binary Search Tree: $\mathcal{O}_{\text{worst}}(\underline{\hspace{2cm}n\hspace{2cm}})$ $\mathcal{O}_{\text{amortized}}(\underline{\hspace{2cm}n\hspace{2cm}})$

بدترین حالت برای یک درخت دو دویی این است که به شکل یک لینکد لیست در آمده باشد که میشود $O(n)$. حالت متوسط هم بدون اضافه کردن محدودیت یا فرض های بیشتر نمیتوان اضافه کرد و $O(n)$ مورد قبول است. اگر فرض دیگری کرده باشید مثل بالانس بودن ... پیچیدگی محاسباتی متناسب با فرض شما ممکن است متفاوت باشد.

(b) Find in an AVL Tree: $\mathcal{O}_{\text{worst}}(\underline{\hspace{2cm}\log n\hspace{2cm}})$ $\mathcal{O}_{\text{amortized}}(\underline{\hspace{2cm}\log n\hspace{2cm}})$

درخت دو دویی AVL همیشه بالانس است و در بدترین حالت و حالت متوسط ارتفاعش $O(\log n)$ است. پس پیچیدگی محاسباتی پیدا کردن هم میشود $O(\log n)$.

۴. [۴۵] فرض کنید Binary Search Tree یک داریم. توضیح دهید خروجی چیست و چه عملیاتی انجام می شود.

```

F(Tree r)
  while r.left ≠ NULL do
    r ← r.left
  end
  print(r.data)
    
```

Finding the minimum element in a binary search tree.

```

F(Tree r)
  while r.right ≠ NULL do
    r ← r.right
  end
  print(r.data)
    
```

Find the maximum element in a binary search tree.

```
F(Tree r, int k)
  if r == null or r.key == k then
    return r
  end
(c) if r.key < k then
    return F(r.right,k)
  else
    return F(r.left,k)
  end
```

Finding an element in a binary search tree.