Основи програмирања

Вежбе 14

Исидора Грујић
isidora@uni.kg.ac.rs

Лазар Илић
iliclazar15@gmail.com

Факултет инжењерских наука
Универзитет у Крагујевцу

децембар 2023.

Садржај

Садржај

Бинарна стабла - *Binary Trees*

- структура података базирана на показивачима (као и уланчане листе)
- сваки члан секвенце или скупа података је смештен у чвор који садржи три показивача
- сваки чвор има 4 поља, резервисана за вредност тренутног члана и поменута три показивача:
  - ***parent*** - показивач на претходни ("родитељски") чвор
  - ***left*** - показивач на следећи чвор са леве стране (*left child*)
  - ***right*** - показивач на следећи чвор са десне стране (*right child*)
- на овај начин настаје структура слична правом стаблу
- могуће је задржати такву структуру да се до било ког чвора може стићи у $\mathcal{O}(\log n)$ времену

Пример 1

Дефиниција класе за чвор бинарног стабла.

Битни термини

- ***root*** - корен стабла - чвор који нема "родитеља"
- потомци и преци (претходници) датог чвора
- ***leaf*** - лист - чвор који нема "потомке"
- ***sub-tree*** - подстабло - може бити лево или десно; било који чвор у стаблу може имати своје лево или десно подстабло. У питању је стабло чији је корен леви или десни потомак датог чвора.

Позиција чворова

Што се тиче позиције чворова у стаблу, разликујемо два основна параметра, а то су:

- **дубина** чвора
  Дубина чвора одређена је бројем ивица ("грана") које се могу уочити на директном путу од корена стабла до датог чвора.
- **висина** чвора
  Висина чвора одређена је бројем ивица ("грана") на директном путу од датог чвора до најдаљег листа стабла.
  висина стабла = висина корена стабла

Уопштено, можемо рећи да је за проналажење неког чвора стабла потребно $\mathcal{O}(h)$ времена, при чему је h висина стабла.

Обилажење стабла

Inorder Traversal Order

За дати чвор X бинарног стабла:

- сви чворови у подстаблу $X.left$ обилазе се пре чвора X
- сви чворови у подстаблу $X.right$ обилазе се након чвора X

Одакле почети обилажење у датом стаблу (или подстаблу)?
Идеја је да се пронађе крајњи леви чвор (лист).

На који начин се одређује следећи чвор који је потребно обићи?

- ако дати чвор има десног потомка, пронаћи крајњи леви чвор у подстаблу чији је корен тај десни потомак
- у супротном, "попети" се у стаблу све док не дође до тога да се дати чвор налази у левом подстаблу тренутног чвора

Каква је временска сложеност ових операција?

Inorder пролазак кроз стабло - Пример 2

```
def inorder(root, array):
    if not root:
        return

    inorder(root.left, array)
    array.append(root)
    inorder(root.right, array)
```

Додавање и брисање чворова

Додавање новог чвора након датог чвора:

- ако дати чвор нема десног потомка, додати нови чвор на то место
- у супротном, поставити га на место левог потомка чвора од ког почиње обилажење подстабла чији је корен десни потомак датог чвора

Брисање чвора:

- тривијалан случај - брисање чвора који представља лист стабла - одсецање чвора
- у супротном, врши се замена места датог чвора и његових претходника све док се дати чвор не нађе на позицији листа, када се врши његово одсецање

Проналажење чвора са одговарајућом вредношћу

Binary Search Tree - Бинарно стабло претраживања

Претрага у бинарним стаблима је врло слична бинарној претрази код сортираних низова.

Потребно је кретати се кроз стабло на одговарајући начин, помоћу показивача.

Пример 3

Операције над бинарним стаблом претраживања.

Балансирање стабла

- у најгорем случају, висина стабла једнака је броју чворова n - небалансирано стабло
- идеја је ограничити висину стабла тако да буде реда величине $\log n$
- тако се постиже циљ да све операције над стаблом имају логаритамску сложеност

Пример 4

Добијање балансираног бинарног стабла претраживања из небалансираног бинарног стабла претраживања.

Поређење - Низови, Уланчане листе, Бинарна стабла

Операције / Структуре података	get_at(i) set_at(i,x)	insert_first(x) delete_first()	insert_last(x) delete_last()	insert_at(i,x) delete_at(i)
Статички низови	$\mathcal{O}(1)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$
Уланчане листе	$\mathcal{O}(n)$	$\mathcal{O}(1)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$
Динамички низови	$\mathcal{O}(1)$	$\mathcal{O}(n)$	$\mathcal{O}(1)_{(a)}$	$\mathcal{O}(n)$
Бинарно стабло (општи случај)	$\mathcal{O}(h)$	$\mathcal{O}(h)$	$\mathcal{O}(h)$	$\mathcal{O}(h)$
Балансирано бинарно стабло	$\mathcal{O}(\log n)$	$\mathcal{O}(\log n)$	$\mathcal{O}(\log n)$	$\mathcal{O}(\log n)$

Садржај

Задаци

Задатке са четрнаестих вежби можете пронаћи на страници предмета.