مراحل زیر را انجام بدهید:

هدف از این تکلیف ارزیابی عملکرد binary search tree میباشد. تکلیف بر اساس چیزیکه خواسته شده انجام شود.

دو فایل زیر را دانلود کنید

names\_frequencies\_2017\_to\_2020\_SORTED.csv

names\_frequencies\_2017\_to\_2020\_SHUFFLED.csv

این مجموعه لیستی از اسامی ست که توسط ایالات متحده منتشر شده است (اداره تامین اجتماعی)

مقادیر در هر سطر از این دو فایل به وسیله ی کاما از هم جدا شدند

x, frequency

frequency در واقع همان تعداد دفعاتی است که نام x در اداره ی تامین اجتماعی ذکر شده است.

با این حال، همانطور که نام فایل نشان می دهد، اسامی در فایل اول بر اساس حروف الفبا مرتب شده اند، در حالی که

فایل دوم به‌طورنامرتب لیست شده اند.

در این تکلیف،ما در Dictionary data structure مقدار key را نام اشخاص و value را همان فرکانس در نظر خواهیم گرفت.

به عبارت دیگر، یک Dictionary به ما خواهد گفت که : «نام X چند بار در ایالات متحده آمریکا در سال های 2017 - 2020 ذکر شده است ؟

2- برای هر کدام از دیتا ستهای بالا باید یک Binary Search Tree از نوع no Balancing بسازید .

data structure باید توسط شما اجرا بشه نه با استفاده از off-the-shelf implementations.

Binary Search Tree در ابتدا باید خالی باشد.

اکنون، برای هر ردیف از مجموعه داده، ورودی نام را به عنوان key و frequency را به عنوان Valueاضافه کنید.

زمان صرف شده برای درج ورودی در هر BST را جداگانه ثبت کنید.

سپس، زمان صرف شده برای جستجوی ورودی را که به تازگی اضافه کرده اید اندازه گیری کنید.

binary search tree دارای فیلدهای اشاره گر چپ و راست هستند، یک اشاره گر والد (اختیاری)،و یک مقدار دیتا.

یک نمونه ازساختار درختی در زیر آورده شده است:

typedef struct tree {

item\_type item; /\* data item \*/

struct tree \*parent; /\* pointer to parent \*/

struct tree \*left; /\* pointer to left child \*/

struct tree \*right; /\* pointer to right child \*/

} tree;

درج در یک درخت باید به صورت زیر انجام شود

insert\_tree(tree \*\*l, item\_type x, tree \*parent)

{

tree \*p; /\* temporary pointer \*/

if (\*l == NULL) {

p = malloc(sizeof(tree)); /\* allocate new node \*/

p->item = x;

p->left = p->right = NULL;

p->parent = parent;

\*l = p; /\* link into parent’s record \*/

return;

}

if (x < (\*l)->item)

insert\_tree(&((\*l)->left), x, \*l);

else

insert\_tree(&((\*l)->right), x, \*l);

}

tree \*search\_tree(tree \*l, item\_type x)

{

if (l == NULL) return(NULL);

if (l->item == x) return(l);

if (x < l->item)

return( search\_tree(l->left, x) );

else

return( search\_tree(l->right, x) );

}

3- نتایج خود را با استفاده از نمودارهای زیر خلاصه کنید:

* زمان ورود اطلاعات در مقابل تعداد رکوردهای BST برای داده های مرتب شده
* زمان جستجو در مقابل تعداد رکوردهای BST برای داده های مرتب شده
* زمان درج در مقابل تعداد رکوردهای BST برای داده های نامرتب زمان
* جستجو در مقابل تعداد رکوردهای BST برای داده های نامرتب

نحوه ی ارسال تکلیف

همه فایل extension فایل را قبل از آپلود به txt تغییر دهید.

فایل کدها به صورت جداگانه باشد( برای هر کدام از دیتاست های داده شده).

هر سورس رو بصورت جدا آپلود کنید.

بصورت زیپ فایل ارسال نکنید.

یک گزارش (سند PDF، حداکثر 5 صفحه) که کد، نمودار،نتایج شما را توضیح می‌دهد.

4- از زبان های C#, C, Python, C++

5- اجراها باید براساس برنامه ها کیفیت بالایی داشته باشند، شرایط خطا و باگ ها را بررسی کنید و

دستورالعمل های استاندارد کدنویسی را در نظر بگیرید، بالای هر کد بنویسید توضیح کد نوشته شود.

6- منابع اگر براساس کدیی از وبسایتها و منابع دیگری باشد نمره صفر منظور میشود.