

Inverted index project

Phase 1

9431052 maryam ebrahimzadeh

نحوه پیاده سازی این پروژه این طور بود که من اول رابط گرافیکی را طراحی کردم بعد در پنل که دارم تابع های خواندن فایل را نوشتم و سپس درخت جستجوی دودویی رو نوشتم و دیباگ کردم سپس کاکمد هارا نوشتم و بعد هم دو درخت دیگر را طراحی کردم. در حین انجام پروژه برا بنوشتم تابع هایی که کل درخت

Trie & TST

را چاپ میکند به مشکل برخورددم. که به پرسیدن نحوه کلی عملکرد از دوستان و سرچ در اینترنت مشکل حل شد.نوشتن تابع

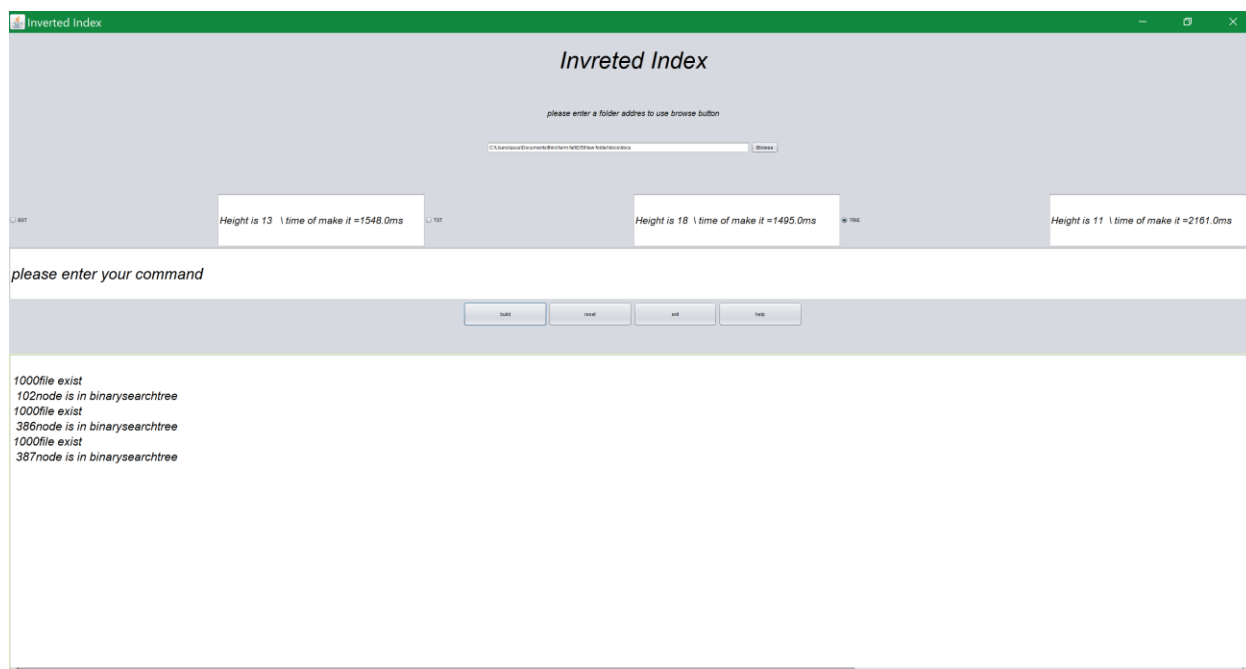
Insert

برای

TST

هم سخت بود من اول چیزی برای ریتزن تابع درنظر نگرفته بودم برای همین خیل کارم سخت بود اما بعدا به تابع ریتزن دارم و بهتر شد.

خروجی کارم در نگاه اول برای 3 درخت این است



مشاهده میشود

	Time of build/ms	Height	Number of nodes
BST	1548	13	102
TST	1495	18	386
TRIE	2161	11	387

که میبینیم درخت

BST

زودتر ساخته شده است. (طبق مشاهدات قبلی البته درخت اول و دوم همزمان بودند و درخت دوم کمی زودتر ساخته میشد) سپس

BST & TRIE

1. TST
2. BST
3. TRIE

و میبینیم که ارتفاع درخت ها چگونه است. ارتفاع درخت اخر با احتساب نود اول است.

پیچیدگی زمانی توابع در خود کد بالای هر تابع نوشته شده است.

	add	delete	search	Traverse
BST	Log(n)	Log(n)	Log(n)	O(n)
TST	O(h)=Log(n)	O(h)=Log(n)	O(h)=Log(n)	O(n)
TRIE	O(h)=Log(n)	O(h)=Log(n)	O(h)=Log(n)	O(n)

برای کلاس لینک لیستی که داریم سه تا تابع سازنده داریم که هم برای ذخیره فایل ها از ان استفاده میکنم هم برای دنو های بچه در درخت سوم هم برای ذخیره کامند ها در استک. بنابراین تابع اضافه و کم کردن هم در این لینک لیست سه تا است.

تماما جدول برحسب میلی ثانیه است.

	Add d1	Delete d1	Search – w york	Search –w maryamebrahimzade Not found	Search –s York anyway times	Search –s maryam anyway university Not found	List -w
BST	2	1	0	0	233	6	206680
TST	6	1	0	0	1584	6	203958
TRIE	8	1	0	0	1665	7	190880