گزارش پروژه ی اول ساختمان داده

محمد امین علیاری - ۹۴۳۱۰۶۶

نحوه ی پیاده سازی

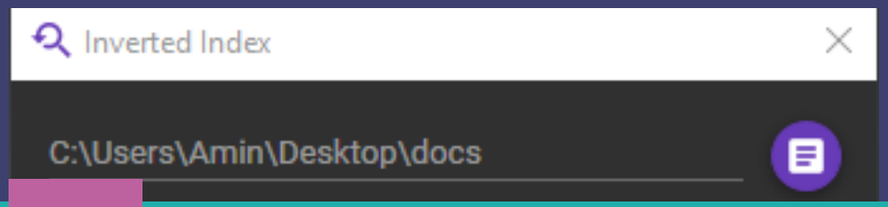
زبان . تکنولوژی . مراحل

برای پیاده سازی پروژه از زبان برنامه نویسی C# و تکنولوژی .Net استفاده شده است.
محیط برنامه نویسی نیز Visual Studio بوده است. در این برنامه تنها استفاده از ساختمان داده های استفاده محدود به System.Collection.List می باشد که عملکردی مانند یک آرایه ی داینامیک مانند Linked List دارد.

همچنین این برنامه دو Command اضافی دارد:

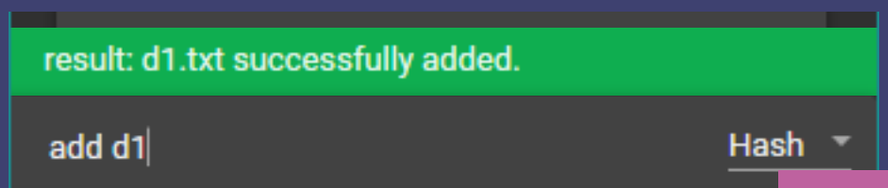
۱. cls (clear) به منظور پاک کردن صفحه
۲. addall به منظور اضافه کردن همه ی فایل های یک دایرکتوری

نحوه ی استفاده



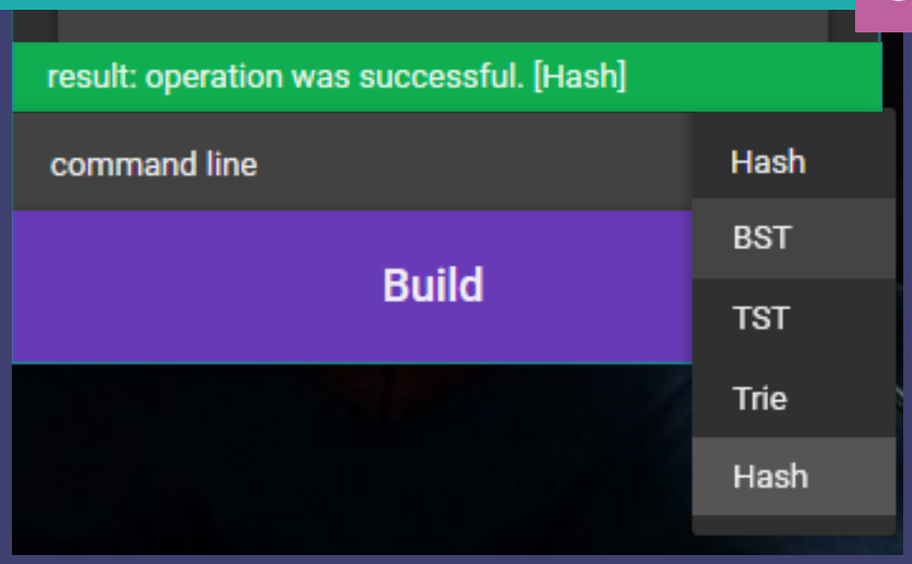
01 انتخاب مسیر دایرکتوری

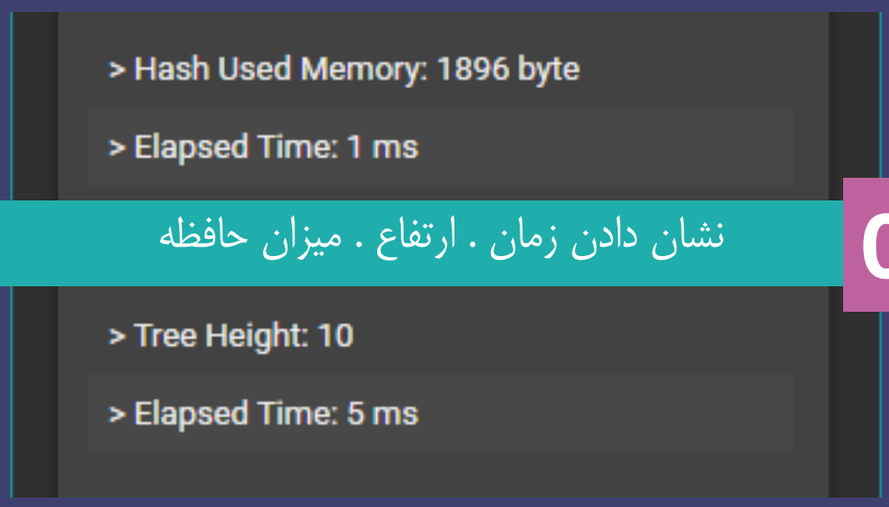
02 لیست کلمات اضافه شده



03 نحوه ی اضافه کردن فایل . پیام موفقیت

04 انتخاب متد . زدن دکمه ی build . پیام موفقیت





05 نشان دادن زمان . ارتفاع . میزان حافظه

ساختمان داده های استفاده شده

- Binary Search Tree (BST)

به گونه ای پیاده سازی شده است که به طور خودکار بالانس شود. یعنی در واقع یک AVL (Self adjusting) Tree

- Ternary Tree (TST)

به گونه ای پیاده سازی شده است که به طور خودکار بالانس شود. یعنی Balanced Tree

- Trie Tree

- Hash map

این ساختمان داده از یک تابع getHash استفاده می کند.
کارکرد آن به این صورت است که hash هر object را گرفته و بر سایز جدول تقسیم می کند.
این سایز متغیر درنظر گرفته شده تا آزمایشات مختلفی قابل انجام دادن باشد.
همچنین برای جلوگیری از Collision بین object ها از تکنیک Chaining با استفاده از یک List استفاده شده است.

مقایسه ی سرعت و حافظه

به ازای ۱۰۰۰ سند موجود در پروژه

Method	کلمه ی world			کلمه ی days			جمله ی days of world		
	Time (ms)	Height	Memory	Time (ms)	Height	Memory	Time (ms)	Height	Memory
BST	19	7	2kb	3	7	2kb	84	7	624b
TST	3	15	11kb	2	15	0	73	15	11kb
Trie	1	10	25kb	3	10	0	62	10	25kb
Hash (128)	6	-	3.5kb	12	-	3.5	79	-	3.5kb
Hash (512)	5	-	4kb	3	-	4kb	59	-	4kb

نتیجه گیری

- Binary Search Tree (BST)

  از دو درخت دیگر جای کمتری میگیرد اما سرعت پایین تری دارد.

- Ternary Tree (TST)

  از BST سریعتر است اما حافظه ی بیشتری مصرف میکند.

- Trie Tree

  از هردو مورد قبلی سریعتر است اما اصلا از لحاظ مصرف حافظه بهینه نیست.

- Hash map

  این نوع ساختمان داده تقریبا هم در سرعت و هم حافظه عملکرد بهتری دارد.
  همچنین استفاده از ۱۲۸ خانه نسبت به ۵۱۲ خانه، حافظه کمتری مصرف میکند اما سرعت پایین تری نیز دارد.
  در کل تکنیک های مختلف هش کردن در این موضوع نقش دارند.

  پیشنهاد من برای بهبود استفاده از TST به جای Linked List در روش Chaining این حافظه است.