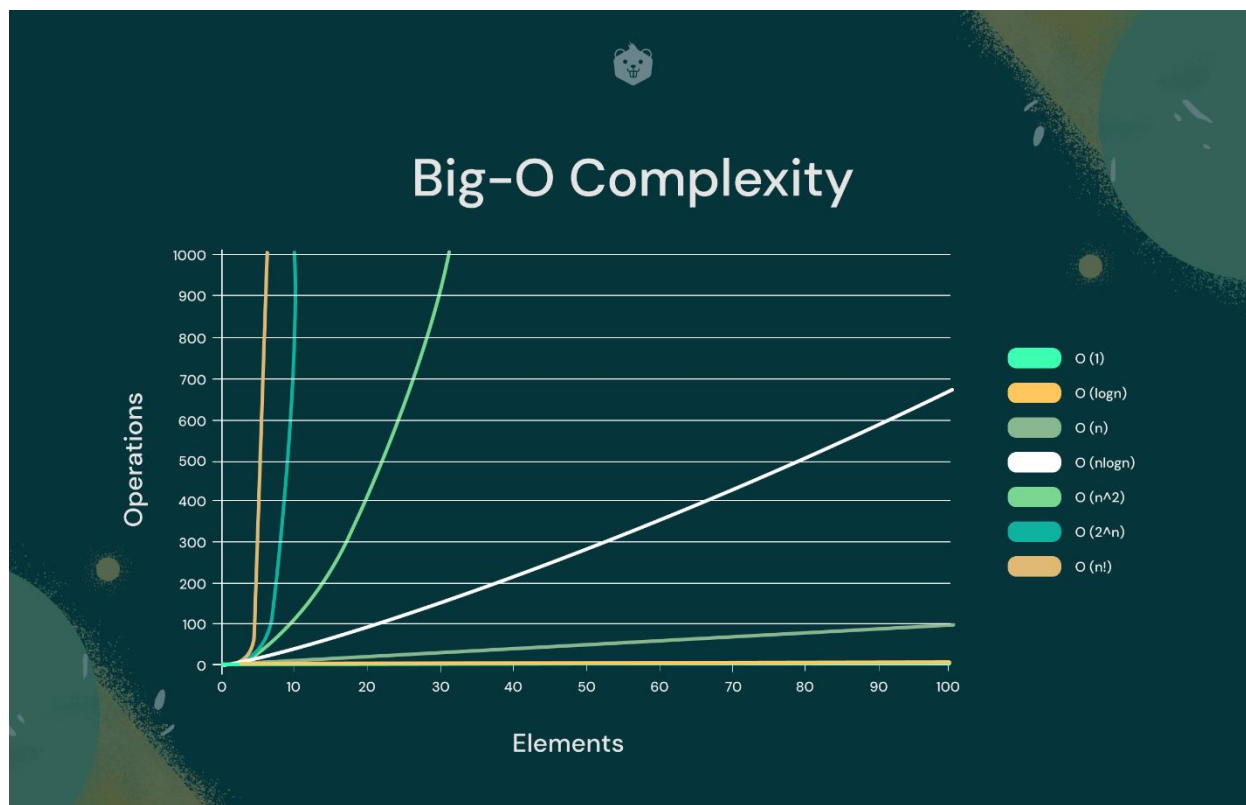




ترم ۱۴۰۲۱

# دانشجوی آزمایشگر



## هدف

هدف از انجام این پروژه آشنایی و تمرین دانشجویان با نحوه تفکر بازگشتی، محاسبه پیچیدگی زمانی کدها و درک مفاهیم این مبحث با به کارگیری آنها در محیط عملی می‌باشد.

## مقدمه

پس از اینکه دانشجویان برنامه‌سازی پیشرفته و ریاضیات گسسته را با سلام و صلوات پاس کردند، با درس ساختمان داده و الگوریتم به عنوان سومین درس خوفناک کارشناسی مواجه شدند! در اولین پروژه این درس، شما به حل یک مسئله ساده، و به پیاده سازی و مقایسه زمان اجرای سه الگوریتم جستجو خواهید پرداخت.

## توضیح

### بخش اول:

در ابتدا باید دو راه حل متفاوت برای مسئله ای که در ادامه توضیح داده خواهد شد بیابید. راه حل اول به صورت بازگشتی و راه حل دوم به صورت غیر بازگشتی. برای هریک از راه حل ها برنامه ای بنویسید و در قسمت مربوطه سوال در کوئرا آپلود کنید. قطعه کدهای ارسالی توسط تست کیس‌ها داوری می‌شوند.

توجه داشته باشید که باید پیچیدگی زمانی تئوری برنامه‌های خود را نیز به همراه راه حل کامل محاسبه کرده و در داکيومنت مربوطه یادداشت کنید.

### مسئله رمزگشایی اعداد:

سارا می‌خواهد برای مکاتبه مسائل محرمانه با دوستش به جای کلمات عادی از حروف به صورت خاص استفاده کند که معنی این حروف براساس قراردادی بین سارا و دوستش مشخص می‌شود. سارا پس از آزمون و خطاهای بسیار متوجه می‌شود که نوشتن حروف یکسان پشت سر هم بسیار زمانبر است؛ پس تصمیم می‌گیرد که برای تولید آنها برنامه ای بنویسد که با داشتن تعداد رشته مربوطه، به تعداد مشخص شده آن را چاپ کند ولی دانش برنامه نویسی ندارد. پس به سارا در نوشتن برنامه‌اش کمک کنید.

برنامه شما در ابتدا آرایه‌ای از اعداد و حروف را به عنوان ورودی دریافت می‌کند و در خروجی آنها را به تعداد خواسته شده چاپ می‌کند. فرمت ورودی‌ها به صورت  $k(\text{string})$  است که رشته داخل علامت "()" دقیقاً  $k$  بار باید تکرار شود که  $k$  حتماً عددی مثبت و بزرگتر از صفر است. توجه داشته باشید که رشته‌ها به هیچ عنوان حاوی چیزی به غیر از حروف کوچک انگلیسی نیستند.

Input	Output
$3(bn2(hj))$	bnhjhbnhjhbnhjhj
$3(a)2(bv)cb$	aaabvbcvcb
$4(o)b$	oooob
$2(f)3(p)$	ffppp
$2(ab2(c2(d)))2(w)$	abccddabccddccddww

به عنوان مثال، ورودی  $3(a)2(bv)cb$  دارای خروجی aaabvbcvcb است. عدد قبل از علامت " ( ) " مشخص کننده تعداد حرف/رشته حروف است. اگر پیش از حرف/رشته حروف، عددی وجود نداشته باشد، یعنی تعداد حرف/رشته حروف بیش از یک نیست.

## بخش دوم:

در این مسئله باید الگوریتم های جست و جوی خطی، جست و جوی دودویی و جست و جوی سه تایی را پیاده سازی کنید. سپس به شیوه ای که در ادامه به آن خواهیم پرداخت، زمان اجرای آن ها را با یکدیگر مقایسه کنید.

### الگوریتم جست و جوی خطی

این الگوریتم، هریک از خانه های آرایه را به ترتیب جست و جو می کند تا آرگومان جست و جو پیدا شود. برای اطلاعات بیشتر به لینک زیر در دانشنامه ویکیپدیا مراجعه کنید.

[https://en.wikipedia.org/wiki/Linear\\_search](https://en.wikipedia.org/wiki/Linear_search)

### الگوریتم جست و جوی دودویی

این الگوریتم روی آرایه های مرتب شده اعمال می شود. بطوری که در هر مرحله آرایه را به دو قسمت مساوی تقسیم کرده و با مقایسه ی مقدار جستجو با مقدار میانی، یکی از دو قسمت را برای ادامه جست و جوی بر می گزیند. این عمل را تا جایی انجام می شود که مقدار مورد نظر پیدا شود. برای اطلاعات بیشتر به لینک زیر در دانشنامه ویکیپدیا مراجعه کنید.

[https://en.wikipedia.org/wiki/Binary\\_search\\_algorithm](https://en.wikipedia.org/wiki/Binary_search_algorithm)

### الگوریتم جست و جوی سه تایی

همانند قبل، این الگوریتم نیز روی آرایه های مرتب شده اعمال می شود. بطوری که در هر مرحله آرایه را به سه قسمت مساوی تقسیم کرده و با مقایسه ی مقدار جست و جو با یکی از مقدار های میانی، یک یا دو قسمت از سه قسمت را برای جست و جو در مراحل بعد بر میگزینیم و این کار را آنقدر تکرار می کنیم تا مقدار قابل جست و جو را پیدا کنیم. برای اطلاعات بیشتر به لینک زیر در دانشنامه ویکیپدیا مراجعه کنید.

[https://en.wikipedia.org/wiki/Ternary\\_search](https://en.wikipedia.org/wiki/Ternary_search)

## مقایسه الگوریتم ها

این مسئله ورودی از سمت کاربر نخواهد داشت، و مراحل آن از طریق تولید کردن آرایه های تصادفی و مقدار های تصادفی برای جستجو های آن انجام خواهد گرفت. پس از پیاده سازی این الگوریتم ها باید در مراحل مختلف آرایه های تصادفی به طول های ۱۰ تا ۱۰۰۰۰ تولید کنید و سپس بشمارید هر یک از این الگوریتم ها برای جستجوی عنصر  $x$  در آن آرایه تصادفی چند عملیات انجام میدهند. سپس به کمک مرتبه اجرای الگوریتم و طول آرایه های بدست آمده، نمودار operation - elements هر یک از الگوریتم ها را جداگانه رسم کنید. (برای این کار میتوانید داده های خود را در اکسل و یا سایر نرم افزار های مشابه وارد کنید و سپس آن را به نمودار تبدیل کنید، و یا با استفاده از کتاب خانه ها و کدنویسی نمودار را ترسیم کنید.)

## محتوای فایل zip ارسالی شما:

۱. قطعه کد، توضیح کارکرد و محاسبه پیچیدگی مربوط به راه حل غیربازگشتی در بخش اول
۲. قطعه کد، توضیح کارکرد و محاسبه پیچیدگی مربوط به راه حل بازگشتی در بخش اول
۳. قطعه کد، و محاسبه پیچیدگی هر سه الگوریتم جستجو
۴. رسم نمودار رشد الگوریتم ها به همراه توضیح جزئیات

فایل فشرده ارسالی شما حتما باید حاوی یک pdf برای ارائه توضیحات، نمودارها و... باشد. در نوشتن داکيومنت دقت کافی داشته باشید، سعی کنید علائم نگارشی را رعایت کنید و از فونت مناسب استفاده کنید.

**توجه ۱: این پروژه ارائه ندارد.**

**توجه ۲: این پروژه نمره امتیازی ندارد.**

## ثبت پروژه

پروژه خود را در فایل فشرده Zip با فرمت زیر بنویسید و آن را در کوئرا ارسال کنید.

DSProject1\_FirstnameLastname\_StudentNumber.zip

## نکات عمومی پروژه‌ها

هرگونه ایده و خلاقیت جدید، تنها با به کارگیری مباحث پروژه و سرفصل‌های مرتبط با این درس و با هماهنگی با دستیاران آموزشی، تا ۲۵ درصد نمره مثبت به همراه خواهد داشت.

استفاده از هرزبان برنامه نویسی بلامانع است.

پروژه تنها به صورت انفرادی قابل انجام بوده و وجود هرگونه تشابه مشکوک بین دو کد یا عدم تسلط به ارائه، شامل جریمه خواهد بود.

تمامی ساختمان‌داده‌های مورد نیاز پروژه باید توسط شما پیاده سازی شوند و استفاده از ساختمان‌داده‌های موجود در کتاب‌خانه‌های زبان برنامه نویسی مورد نظر شما شامل نمره نخواهد بود.

رعایت اصول شیء گرایی و آموزه‌های درس برنامه‌سازی پیشرفته، نام گذاری صحیح، کامنت نویسی مناسب، رعایت دندانه گذاری، و خوانا بودن کد الزامی است.