

Subject:

Year. Month. Date.

Sa	Su	Mo	Tu	We	Th	Fr
----	----	----	----	----	----	----

لaptop

دروس معلوماتية

Data Structures and Algorithms

بيانات وstructures

الstruktures وalgorithms

بيانات وstructures

١٤٠٥ نسخ

بيانات وstructures

Subject:

Date:

ساعت

List و Array - A جمله‌هایی در زیر

آرایه (Array) و لست (List) هم‌درباری ذخیره صحیح نهایی از عناصر استفاده کنندگان
کمیابی داشتند و وجود نداشتند.

از اندازه (size) :

آرایه (Array) : آرایه اندازه آرایه در زمینه ایجاد آن تأبیت است و نیز نوآن آن را بین تغیرات
نمی‌باشد و می‌تواند که می خواهد صحیح نباشد و آرایه ذخیره نشود. لست (List) نوآن
لست (List) نوآن را نیز پیدا می‌کند و می‌تواند در طول زمان تغییر کند و نوآن عناصر
را بین لیست اختصاص دهد. خنثی نیز درون آن نوآن پرورش خواهد داشت.

2 نوع داده: (Data Type)

آرایه (Array) و دریشتر، بازن‌هایی بینایه نویس آرایه های بینایه نویس نوام داده صحیح باشند
(می‌توانند اعداد صحیح را هم رسانند)

لست (List) : لیست های مجموعه ای از عناصر از انواع داده مختلف را در خود دارند.
دسته ای این دو گروه بازن‌هایی رسانی نمی‌کنند و عکس ای لست های نیز در
بنیان زبان محدود نیستند.

3 عمل، (Performance)

آرایه (Array) : به دلیل اندازه ثابت و نوع داده صحیح دسترسی به عنصر آرایه سریع و ساده
است. لست آرایه های دیگر عملیاتی کمتری از دسترسی دسترسی عناصر دارند (ماشین صنعتی
سازنده) این اسباب ترکیبی

لست (List) : به دلیل اندازه بیوای درج و حذف عناصر در لیست می توانند نهاده از آرایه های

4 حافظه (Memory)

آرایه (Array) : آرایه های بحد ذاتی پیوسته در حافظه ذخیره می‌شوند و این امر باعث می‌شود
دسترسی به عنصر از های سریع نباشد.

لست (List) : لیست های مجموعه ای به صورت پیوسته در حافظه ذخیره می‌شوند و عناصر آن های عکس

لست در مکان های مختلف های فضای پردازشی باشند این اصول نواده باشد که نیاز به تغییر داشته باشند

۲- ویرگول (f caters)

آرایه ها (Array): آرایه های ساخته شده برای ذخیره و پر کردن یک مجموعه ای از داده ها

لست (List): لست های ساخته شده برای ذخیره و پر کردن یک مجموعه ای از داده ها که ممکن است ترتیب داشته باشد

لست های خوب جستجو و مرتب کردن عنصر

مثال :

فهرستی خواصی فقرات در داشت که زیرا ذخیره سازی آنها دشوار است از قبل مخفف باشند
لست دارای آن را بسیار ممکن است از آن بعد از فهرست مخفف نباشد و بخواهد که فهرست را با آن تبدیل
افراز کنید استفاده از لست های از پرایم

3- دیکشنری (Dictionary):

در پایتون لایک دیکشنری (dict) یک ساختار داده ای است که برای ذخیره مسازی
جفت های کلید و مقدار استفاده می کند که ممکن است دیکشنری صفحه بوده باشد و دیکشنری
مرتب سازی شود این ساختار داده ای سیار افعلا فیزیکی نداشته باشد ذخیره مسازی
و بازیابی داده های بمحوریت کار آنها مستغل نماید

درینه های اصلی دیکشنری فر:

- ۱- کلید منحصر به فرد هستند که باید یکتا باشد، آنها کلیدی تقدیر می شوند وارد شوند
سقراط جدید حافظه زیر صفت از قبیل جی کلود
- ۲- درست در نسخه های جدید پایتون، پایتون لزگ و آن به بعد دیکشنری های ترسیم درج و احتفاظ می کند
- ۳- انعطاف پذیری که کلیدها و مقادیری را که نیاز است ذخیره نموده ای از دیکشنری های باید بازخوانی شوند
داده ای باشند (ما ملید های باید از نوع داده ای تغییر ناید) (صلح و تغیرها اعوام دیتا می باشند)
ایجاد دیکشنری داده های بسیار به مقادیر ۳ افزودن با به روز رسانی مقادیر که حفظ شده باشند
- ۴- سفارت (کی برسی) وجود کلید

Date:

Subject:

روشن های مخصوص دیکشنری:
 کیس (KEY): این ها کلید های ابجع اند که کوئل (VALUES) لیست مقادیر ابجع را در
 کوئل (List) لیست جنگ های کلید مقادیر را برمیگردانند

List و Tuple C

لیست های اصلی بین Tuple و List در پایتون بیشتر زیر آمده اند

1- تغییرپذیری:

List: لیست های تغییرپذیر است که میتواند عناصر را تغییر دهند اما نه تنها اینها
 Tuple: تابل های تغییر ناپذیر هستند پس از آن که یک تابل میتوانند متغیر باشند و تغییر داده افراطی با جنگ های دیگر

2- خواص:

List: لیست های با اتفاق (های برآمده اند) تغییرپذیرند

Tuple: تابل های با اتفاق (دارند) تغییرپذیرند

3- عملیات:

List: تابل های به دلیل تغییر تابل بودن ممکن است که از لیست های سریع نه صندوق پیشوند میتوانند تابل های

تغییرپذیری داشته باشند

List: لیست های به دلیل نیاز به تغییرات اعمانی عملیات کسر نه صندوق

4- حافظه:

Tuple: تابل های به دلیل تغییر تابل بودن ممکن است که حافظه کمتر باشند نسبت به لیست های صندوق

List: لیست های صندوق اعمانی تغییرات اضافی برای تغییرات اضافی

حافظه کمتر (List) صندوق است

5- متد ها:

remove append(): این متد عناصر را از دستگاری حذف میکند

(insert(t, index, count)) دارنده این متد دستگاری جنگجوی ایجاد میکند

Tuple: تابل های صندوقی نعمت دارند سمعت ایجاد جنگجوی ایجاد میکنند

۱- میراث (Use Cases):

List: برای مجموعه های از داده که نیاز به تغییر را دارند (مانند اضافه، حذف، ...). هنوز کاردن نباشد و وزیر امنیت
متوجه مناسب نمایند.

Tuple: برای مجموعه های از داده که نیاز به تغییر نداشته باشند (مانند مختصات، جمله هایی، ...).
البته این مناسب نمایند مخصوصاً بسیار کلید در دیکشنری های استفاده شوند. برای
کلید های ناید تغییر ناین بود.

۲- SET در Python چه باید حذف داده های تکراری استفاده فرمود؟

استفاده از SET در پایتون برای حذف داده های تکراری به دلیل زیر است
۱- منور است.

۱- منور است که بعد از تغییر تبعیع تغییر مختصر شود. در آخرین زمان
به این معنی است که SET عنصر تکراری را نیست. SET اضلاع می توانند دو عدد از هر
کدام از آنها در یک جایی خواهد صاف شد.

۲- مثلاً در پایتون می توانی (A): عملیات های اضافه و حذف و جستجوی عنصر
در SET بدلیل استفاده از جدول در دیکشنری (Dictionary) بیان سریع نمایند.
این ویژگی های بعده از شرکت کردن حذف داده های تکراری بسیار وسیع باشد.

۳- سادگی در پیاده سازی: تبدیل کردن سمت برابر SET برای حذف کار ای (Delete operation) و
خوانا است. فقط کافیست لیست خود را به شکل کوچک نمایند و به این ترتیب تکراری های از طرف خود حذف شوند.

۴- عدم تعجب به ترتیب: که نام ترتیب استدای این سخن ام از ترتیب اعماق در هنام حذف کار ای
نمایند. این ویژگی کار ای نو از نزد در بخش سفارشی کار کند. با این

استفاده از Q و S بدی منف داده طبیعی در بایتون به خاطر خاصیت صنایع
فرم بودن علاوه بر سرتی بالای عملیاتی استفاده می‌نماییم، بنابراین آنها بعمل
کرد راهنمایی و ساده شناخته می‌شود.

Queue و Stack

کافایت اصلی Q و S دسته اضافه و حذف عنصر (LIFO) است.

$Stack$ (لیست): آخرین عضوی که وارد $Stack$ شد اولین عضوی است که خارج از $Stack$ می‌شود.

است در پردازش مورد نظر است که هر کدامیکی از اخیر بازی باید

آخرین اولین برجای باید

آخر داشتند و دو عملیات اصلی دارند

$stack$: اضافه بردن به عنوان $push$. حذف از آخر $stack$ به عنوان pop .

کاربردها:

1. بروزگر غلطی پرانتزها در عبارات ریاضی ۲. پیاده‌سازی

3. قابلیت توابع در برنامه‌نویسی (مانند $allStack$)

Depth-First-Search

: (صف) Queue

Queue, اولین عضوی از Q : FIFO(first_in, first_out)

عنصری است که خارج از Q خواهد شد (صف زانویی). بی را اول ایند اول ایند

MICRO

لیست دفعی: دفعه داده، اعمق و خود

Queue: اضافه درون بعده باید انتظاری ENQUEUE

Queue: حذف در پیش از این باید انتظاری DEQUEUE

لیست دفعی: اضافه در خواسته دارد که سینه (ماشین) را باید حذف کنی (جایی)

BF

الگوریتمی: جستجوی پیشین (Search first) اولین عاقلاً

لیست دفعی: اضافه در خواسته دارد که سینه (ماشین) را باید حذف کنی (جایی)

لیست دفعی: جستجوی عمق در پیش از داده داده

لیست دفعی: Queue: اضافه در خواسته دارد که سینه (ماشین) را باید حذف کنی

لیست دفعی: کارهای دستیع عالم

چیست و چه کاربرد دارد؟ HashTable

جدول هش (HashTable): مساخته را در داده است که برای ذخیره و بازنگری داده

با استفاده از (keys) می توانیم درین مساخته همکنی بسیار کریم کرد

(هش آندس در این آن سیستمی خود نهایان آدرس (address) را ایجاد می کند (hash function))

Subject:

ایجاد دلست با استفاده از تابع هش (Hash Function) ایجاد شود

لذا تابع هش (Hash Function) برای ذخیره داده های را در جدول هش ذخیره می کند اینکه می بینیم که تابع هش (Hash Function) که درینجا مذکور شده کیفیت عددی داشته باشد (Index) بجزی برداشتن داده های داشته باشد و سرعت اجرا داشته باشد.

در اینجا داده دار در درون مجموعه و توزیع تابع هش ذخیره می کند و
بازیابی (Data Recovery) بازیابی داده دوباره نماید، این تابع هش که داده های تابع هش داشت همان داده های خواهد بود
(Hash Function) از دو نوعی می باشد دو صریح طرز (Explicit) و داده های بازیابی نه
چهار جدول هش استفاده می کنیم:

جدول هش به دلیل سرتاسری بالا (جستجو در جدول هش برای داده های داشته باشند) زمان
زمان (Time) عملیات ها (1) است یعنی بهتر از داده های دوستی نیست و ثابت است این در
مقایسه با جداول دارای حمل (Average Case Cost) $O(n^2)$ یا بیش از $O(n^2)$ بگیرد
محض و درست است:

تصادم ها (Collisions): چنانچه این تابع هش برای دو دیده مختلف همان داده های
محاسبه می کند پس در تابع هش ممکن است دو داده های مغایر باشند که نتیجه آن (Collision)
یا آدرس دهنده باشند (Open addressing) صریحت است وجود تصادم های عاده همان
بینجود درج و حذف افراد از لیست است

لذا تابع هش: انتخاب تابع هش مناسب بسیار محتم است که تابع هش فوباید توسعه یافته
از اینجا از درج طرف هش ای کنترل از تعداد تصادم ها که می تواند از

کاربرد داشته باشد:

جدول هش در بسیاری از این بروها استفاده می شود از جمله:

پیاده سازی دیکشنری های مجموعه های دارای مجموعه های بزرگ نیزیاری از زبان های برنامه
نویسی از جدول هش برای پیاده سازی دیکشنری های مجموعه های از اینجا می شود

آنچه می دانیم در اینجا جدول هش برای ذخیره داده های پرستاخ در نظر گرفته شده
است که در اینجا باید داده های پرستاخ برای همان رفع نظر کرد

Subject:

Date:

اصلی کری: درین آندر یعنی مای روت و می خواهد اینجا می خواهد

دستورهای جستجو و نوودهای جستجو از جمله دستورهای ذخیره کاری ایجاد

دستگاه های ایجاد سیم پلیسی و میزبانی پایانه داده چند دستورهای ایجاد سیم پلیسی

برای ایجاد سیم پلیسی برای ایجاد داده ذخیره کاری ایجاد سیم پلیسی

B-Tree, Binary Tree - 6

تفاوت های ایجادی بین درخت دودوی (Binary Tree) و درخت بیانی (Balanced Tree)

درخت دودوی (Binary Tree):

ساختار: دفترچه حداقل دو فرزند اولی فرزند هیچ عکس فرزند را ندارد

نامهی: درین درخت دودوی جستجو سمت چپ صفاتی که از این دارند

علق: می تواند بسیار عالی شود به خصوص آرایه های با بلوک نامحدود و از دستور

مسکونی داشت ایده ای (دشت میانی) جستجو درجه و نصف درجه (log n)

(با محض عده ایجاد تردد کمالی)

خضای ذخیره‌سازی؛ خضای ذخیره‌سازی برای سیستم نسبتاً بسیار است

اسفراوه: برای ذخیره‌سازی صریب شده در حافظه اصلی (RAM) می‌باشد که برای دسترسی ها و سیستم‌های ذخیره‌سازی داده‌های پردازش کر نیاز به خواندن و نوشتار کند. داده‌های ذخیره‌سازی این‌طور نیست

ویکیلر: ذخیره‌های دودویی ناچیغی‌تر از عکلله را به طور کامل توجه نمی‌کنند
(B-Tree در برابر حالت)

درخت B-tree

ساختار ذخیره‌سازی می‌باشد در ذخیره‌سازی معمولی این ساختار می‌تواند جزوی فرزند داشته باشد (پسر ازدواج) برای اندازه درجه (order) ذخیره‌سازی B-tree مخصوص شود

نیز می‌تواند برای ساختار می‌باشد که از داده‌های ذخیره‌سازی برای ذخیره‌سازی معمولی قراردارند عقق درخت B-tree معمولی (است حتی برای مجموعه‌های داده‌برگ این بدلیل داشتند که از زیادی فرزند در آن داشته باشد عکلله را:

جستجو: درجه ذخیره‌سازی در زمان $O(\log n)$ انجام می‌شود که این امر بدلیل معلمات ذخیره‌سازی مجموعه‌های داده‌برگ نیز نیست حقیقت است

خضای ذخیره‌سازی؛ خضای ذخیره‌سازی برای سیستم بیشتر از ذخیره دودویی است ژریه‌های می‌توانند جزوی داده‌برگ نباشند و به فرزندان داشته باشند

Subject:

Date:

استفاده: برای ذخیره سیستم داده ها در دیسک (با حافظه جانبی) بسیار مناسب است
در دینامیک مانند سیستم های فایل و سیستم های ذخیره سازی داده های پردازش استفاده

نمایان: مدل‌گرد خوب برای خودگذشتگی داده های دینامیکی دارد. دلیل نمایان
بودن و دسترسی به بلوک های پردازش است. داده های ممکن است از هر کدام یک عدد عامل در
درجه دو باشد، (حتی برای داده های زانفلتم) پایه ای ربط ندارند.

بلوک خالص درخت B-tree برای ذخیره سازی و بازنگری داده های پردازش است.

بنیادینه است در جایی که درخت دودوی برای ذخیره داده های خودگذشتگی است، هر داده های دینامیکی است.

Graph Data Structures - H برای شبکه های اجتماعی استفاده معمولی

ساختار داده ای ساخت (Graph Data Structure) به دلیل ویژگی هایی که در شبکه های اجتماعی
برقرار است برای شبکه های اجتماعی بسیار مناسب است. بودجه ای این داده های شبکه های اجتماعی استفاده
از گراف در شبکه های اجتماعی انجام

صلح سازی وابط: در شبکه های اجتماعی افراد (اویا احادیث صفات) که به هم وصل شده اند وابط
بینواران های (Nodes)، مفهوم که همچنان که بود وابط بین این افراد وابطی که از
همشون و دنبالهایشان و عضویت در یک گروه و لاین گروه وغیره به عنوان لایل (Edges)
نگاه می‌دارند. گراف های بسطه، طبق این وابط را می‌توان بازیابی کرد.

تحلیل شبکه های پیچیده:

محاسبه های صعبه گرفتار های اعلان (محاسبه های صعبه های افراد هم که متن ایجاد شده های
بین این افراد را می‌دانند) امثل استفاده از الگوریتم های جستجوی محدودیتی (DFS یا BFS) یا
یا پیشنهاد داده های بین اسما می‌باشد. وابط موجود (گل) دوستان (روستان) هایها) بسیار ساده است

© MICRO

شناختی جماعت: آنکه چه مجموعی از چند جماعت (Community Detection) یعنی
گراف هایی که از گروه های مجزا ایجاد شده اند که درون گروه اجتماعی و متمایز اند که بین این گروه های اجتماعی
بینویسی تبلیغات پیشنهاد می کنند و تحلیل رفتار و تحریر برای این گروه های اجتماعی

مختصات سنی (Centrality measures) مختصات سنی یعنی مختصاتی که
معنی طبقه ای از افراد با روابط را در شبکه نشان می کنند که این افراد
شناختی افراد با تقدیر زیادی این محظوظین (صفحه های)

انقطاف پذیری و میزان پذیری : میزان پذیری این افراد که بین افرادی که با آنها ارتباط
نماید داشته باشند و میزان پذیری این افراد که بین افرادی که آنها را بازدید نمایند
باشد که این ارتباطات میان افرادی که میزبانند این افراد را این افراد را این افراد را
لهم عظیم لذت دارند و این افراد را آنقدر خوبی دارند که بین افرادی که با آنها ارتباط نداشند

فعالیت اجتماعات پیشیمه ها :-

ویژگی های (Attributes) : این اثبات ویژگی های مختلف را که در هر گروه می باشد
بر این شکل می نشانند اما این اثبات هایی هستند که (نام، سن، جنگ...) را به افرادی افراد (و احوالات)
مخصوصی می دهند و این احوالات مخصوص (جنسیت، میانگین سنی...) می سیال ها اتفاق فرازند

گراف های جهت دار و وزن دار : این از گروه های جهت دار (Directed Graphs) می باشند
که این گروه های روابطی را که این افراد بین افرادی دارند که این افراد را در شبکه (Weighted Graphs)
بین افراد بین افراد نشان دهند (جهت روابط این افراد تفاوت دارد) ایجاد کردند

کاربردهای اصلی :

پیشنهاد دوستان و محتاطه بر این افراد روابط صدیقو و آنها را از این گروه های می توانند
دوستار و دشمنی داشتند و میتوانند روابط را که با افراد برای این گروه های اتفاق افتد

صدیقیت چشم خصوصی : گراف های این اندیشه ای که این افراد را از افراد دیگر می خواهند

Subject:

Date:

خصوصی در کسبهای اجتماعی معمد اسلام قدر نمودند.

نتیجه ی دلیل و رفتار صنعت: با تحلیل آنکه های رفتاری و روابط در تأثیر عوامل

حساب های کاربری جنس فناوری های اینچه و سایر رفتارهای صنعت رسانسایر مورد

بازخواستهای اینکه در مدل تعادل سازی روابط پیشیده تحلیل شده های اجتماعی

خطای پذیری و انعطاف پذیری برای اینکه در مدل سازی و صنعت کسبهای اجتماعی اینکه بسیار قدر تغییر میزند

۱. Dynamic Programming هر دفعه مسائل پیشیده را برداشت

بنزناهه فیزیک پویا به دلیل رویکرد دینامیک و مینیمیمیت سازی آن در حل مسائل پیشیده که روبرو دارد این روش باعث است مسائل بزرگتر و حل صعب در هر زیر مسئله مطلع کردن باشد مسائل

کمتری جلوگیری می کند و علاوه بر این فرآیند زیر مسائل دسته بندی شده، حفظ احتمال

نماینی برای از داشتن ناپردازی می کند

بعینه سازی زمان و بازخواست این نتایج زیر مسائل حل مسئله ای را ممکن می کند

محبود آنها جلوگیری می کند لیکن این امر به ویژه در مسائل که تعداد زیادی زیر مسئله

دارد این دارند صندوق های صرف جویی طبقی نموده برای این

لرز مسائل پیشیده زمان از نتایج های هجده جمله ای که اینجا پیدا شدند

Subject:

Year

Month

Date

()

بیانیه سازی حافظه (در بین موارد) که درین صورت برنامه نویس

میتواند مکارهای حافظه نیز را نسبت به روش های دیگر نیاز داشته باشد اما در صورتی

که روش های بسیار زیادی وجود دارند با جلوگیری از محااسبات مدل این می توانند درخواست

به صرف درین در حافظه نداشته باشد

ساختار باخته و سازمان باخته بدانه نویس پویا کنید و روش ساختار باخته و سازمان

باخته بیکاری حل مسائل پیچیده را که می دهد این روکیده دستیابی و درستی در آنرا در

می تند و احتفال خطا را کاهش می دهد

که رید در انواع مسائل این روش در حلیق عیین از مسائل از جمله مسائل بیانیه

محضی در این مسائل ترسیمات و مسائل مربوط به رئیس ها در حل مسائل های متعددی

از کاربرد آن در الگوریتم های سیستمی اکثر بینهایی برخاسته و نیزی تکلید وجود دارد

به حل رفته صه بدانه نویس پویا نیست مسائل پیچیده به زیر مسائل لرچتر حل

بینهای جلوگیری از محااسبات مدل این به اکثر راه حل های ۷، ۸، ۹، ۱۰ مقابل

فهمی پردازد

چیز و چهار روش اوریتم های پیش فروخته است (همی) شود؟ Recursion

بازگشت (Recursion) پس مفهوم در بین اند نویس علوم نام ہیوندی است لہ در آن پس
کابع خود را به طور مستقیم یا غیره مستقیم فرخوانی کی کہ بہ عبارت دیکھ کابع برائی حل مسئلہ
آن را به صائل کو جعل کرو سپاہ تفہیم کرنے وہ فرخوانی خود برائی حل این
مسئلہ لوحیت بہ صوب خنایی حیرت
چراز بازگشت استفادہ ہمی شود؟

لخوانایی و سازی: در بین صائل استفادہ لازی بازگشت باعث ہی تعدد بہ رخواں

کہ جو کادہ ترین روش اوریتم های کا، نئی جائیداد بہ طور مستقیم، تعدادی ریاضی یا متعلق
مسئلہ پیروی کرنے

۲۔ حل صائل پیچیدہ بازگشت بہ خوب برائی حل صائل کو اسرا ختار دارہ ای بازگشت

کارند مانند درخت ها و لاف ها احتساب اسست این مسائل اغلب با استفادہ اوریتم های

کاری (Iterative) پیچیدہ تر مندرجہ

۳۔ تجزیہ صائل بزرگ بہ صائل کو جیسا، بازگشت اعلان تجزیہ کی مسئلہ بزرگ بہ تجزیہ

مسئلہ (و جیسا و قابل صیریت را فر اعمرا نہ این روئی بہ فضیل) در اوریتم های تفہیم و حل

(conquer divide and conquer)

۴۔ اوریتم های پیش فروختہ در بین اوریتم های پیش فروختہ کا نام صبا سازی سریع

سریع سازی ادغام (Merge sort) و بعدی عمل اول (Quicksort)

ا، بازگشتی استفاده از شور (First Search Depth-First)

مثال ساده: تابع صاحب فاکتوریل باید در صحیح متبعت باشد
def factorial(n):
 if n == 1:
 return 1
 else:
 return n * factorial(n-1)

در این تابع factorial باید صاحب فاکتوریل خود را با فراخوانی مجدد کنند
برای در این حالت این فاکتوریل از مردانه من مدد نمی‌شود فخرانی خواهد بود
با اینها محدود خواست:

1. خطای پیته (Stack overflow): هر چند تابع بازگشتی فراخوانی می‌شود برای خود

در پیته فراخوانی (call stack) ایجاد می‌شود آنرا عقیل بازگشت خوبی برای مطالعه و مطالعه می‌شود

2. جھینه سازی (Circular dependency): الگوریتم های بازگشتی معمول است بر دلیل فراخوانی های صدر تابع کارایی بسیار

سبک سبک روشن کاری کنند و این طبقه نایابی کنند با اتفاقاً از این طبقه ۵۰٪ی مانند عذری بازگشتی

memorization, elimination of recursion (elimination of recursion)

نتیجه کیری: بازگشتی این روش را در زمان فریز اسکالار اعلان حل صائل پیشیه

۱.۶ استفاده از روکاری (Memoization): ساده و خلاصه اهمیت آن این حالت باید به صورت دوستی

ویسی های مربوط به آن نداشتم اما در درست از نوع از روکاری های جھینه سازی استفاده می‌شود