

Data Structures and algorithms

Instructor : Dr.Amin Eskandari

Head-Ta's : Hamid Namjoo manesh , Amir Hossein Hemati , Ali Mojahed
Ta's : Amir Mohammad Asadjoo , Padina razmjooi , Armin Janfaza , Alireza Heydari , Mahdi Torabi ,
Maryam Ghorbani , Fatemeh Keshavarz , MohammadReza Fasli , Abolfazl Tadrissi

Week 07 Questions

سوالات تکالیف هفته ۷

سوال اول :

در مورد جدول‌های درهم‌سازی (Hash Tables) به پرسش‌های زیر پاسخ دهید:

الف) هدف اصلی استفاده از درهم‌سازی چیست؟

ب) تابع درهم‌سازی خوب باید چه ویژگی‌هایی داشته باشد؟ دو مورد نام ببرید.

ج) منظور از برخورد (Collision) چیست و چرا وقوع آن اجتناب‌ناپذیر است؟

سوال دوم :

شما به تازگی به عنوان کارآموز به یک سازمان مخفی تحلیل داده‌ها ملحق شده‌اید.

این سازمان برای نگهداری شناسه‌های محرمانه‌ی مأموران خود از یک سیستم درهم‌سازی (Hash Table) استفاده می‌کند.

در اولین مأموریت، از شما خواسته شده است شناسه‌های زیر را به ترتیب ورود مأموران در سیستم ثبت کنید:

$$A = [67, 13, 49, 24, 40, 33, 58]$$

مشخصات سیستم به شرح زیر است:

- اندازه‌ی جدول درهم‌سازی: ۹
- تابع درهم‌سازی:

$$h(k) = (11k + 4) \bmod 9$$

- در صورت بروز تداخل بین شناسه‌ها، سازمان از روش زنجیره‌ای (Chaining) استفاده می‌کند و شناسه‌های جدید در انتهای زنجیره قرار می‌گیرند.

وظیفه‌ی شما:

پس از ثبت همه‌ی شناسه‌ها، شکل نهایی جدول درهم‌سازی را رسم کنید تا مسئول امنیت سیستم بتواند وضعیت ذخیره‌سازی داده‌ها را بررسی کند.

سوال سوم :

شما به‌عنوان کارآموز واحد نرم‌افزار در یک شرکت فروش آنلاین مشغول به کار هستید. این شرکت برای نگهداری شناسه‌ی عددی سفارش‌ها از یک Hash Table استفاده می‌کند.

مشخصات Hash Table به صورت زیر است:

- ظرفیت جدول : 8
- کلیدها عدد صحیح هستند
- تابع درهم‌سازی:

$$h(x) = x \bmod 8$$

- روش رفع برخورد : Linear Probing
- در صورت برخورد، خانه‌های بعدی به‌ترتیب بررسی می‌شوند. (wrapping مجاز است)
- حذف انجام نمی‌شود

شرکت قصد دارد شناسه‌های زیر را به‌ترتیب در جدول درج کند:

45 , 64 , 10 , 36 , 27 , 19

الف) جدول نهایی Hash Table را پس از درج همه‌ی کلیدها رسم کنید.

ب) مشخص کنید کدام کلیدها باعث برخورد شده‌اند و چند بار برخورد رخ داده است.

ج) اگر از روش Chaining به جای Linear Probing استفاده می‌شد، آیا تعداد برخوردها کمتر می‌شد یا بیشتر؟ پاسخ خود را به صورت مفهومی توضیح دهید.

سوال چهارم :

یک رستوران آنلاین می‌خواهد سفارش‌های مشتریان را به صورت **سریع و مقیاس‌پذیر** ذخیره کند تا بتواند:

- سفارش جدید اضافه کند
 - سفارش‌ها را دسته‌بندی کند. بر اساس نوع غذا (مثلاً pizza, burger, salad)
 - اگر تعداد سفارش‌ها از ظرفیت اولیه جدول بیشتر شود، جدول خودکار **دو برابر** شود
- برای این منظور، شما باید یک **Hash Table** با **Chaining** و **آرایه پویا** در پایتون پیاده‌سازی کنید.

مشخصات جدول سفارش

1. ظرفیت اولیه: ۵

2. تابع هش رشته‌ای:

$$\text{hash(key)} = \text{sum}(\text{ASCII}(\text{char}) \text{ for char in key}) \% \text{tableSize}$$

3. برخوردها با **Chaining** مدیریت شوند

4. هر بار که تعداد سفارش‌ها $\leq$ اندازه جدول:

○ جدول دو برابر شود

○ همه سفارش‌ها دوباره هش شوند (Rehash)

5. امکان **سفارش‌های تکراری** وجود دارد

نمونه‌های ورودی و خروجی

Input 1

pizza, burger, salad

Output 1

0 خانه: []

1 خانه: []

2 خانه: ['burger', 'salad']

3 خانه: ['pizza']

4 خانه: []

سوال پنجم :

در یک پلتفرم ثبت گزارش‌های روزانه، کاربران می‌توانند رویدادهای مهمی مثل خطاها، هشدارها یا پیام‌های سیستمی را ثبت کنند. هر رویداد با یک **برچسب متنی کوتاه** (مثل login، crash، payment و ...) شناخته می‌شود. برای ذخیره و مدیریت سریع این برچسب‌ها، تصمیم گرفته شده است که به جای استفاده از ساختارهای آماده، یک **جدول درهم‌سازی (Hash Table)** ساده اما مقیاس‌پذیر طراحی و پیاده‌سازی شود تا مفاهیم پایهای ساختمان داده‌ها به صورت عملی تمرین شوند.

مشخصات جدول درهم‌سازی

ساختاری که باید پیاده‌سازی شود، دارای ویژگی‌های زیر است:

1. ظرفیت اولیه‌ی جدول برابر ۵ است.
2. برای رفع برخورد (Collision) از روش **زنجیره‌ای با لیست** استفاده می‌شود.
3. ترتیب ذخیره عناصر در هر زنجیره، **مطابق ترتیب ورود** است.
4. تابع درهم‌سازی برای رشته‌ها به صورت زیر تعریف می‌شود:

$$\text{hash}(\text{key}) = (\sum \text{ASCII}(\text{key})) \bmod \text{tableSize}$$

اگر پس از اجرای یک عملیات درج (insert) تعداد عناصر جدول برابر یا بیشتر از ظرفیت جدول شود:

- ظرفیت جدول باید دو برابر شود
- تمام عناصر قبلی باید با اندازه‌ی جدید جدول دوباره درهم‌سازی (Rehash) شوند
- ورود برچسب‌های تکراری مجاز است و هر بار به‌عنوان یک عنصر جدید ذخیره می‌شود.

بخش اول) پیاده‌سازی

یک کلاس در پایتون طراحی کنید که:

- از آرایه پویا برای نگهداری جدول هش استفاده کند
- عملیات‌های زیر را پشتیبانی کند:

insert(key) ○

rehash() ○

display() ○

بخش دوم) پس از پیاده‌سازی، دنباله‌ی زیر از برچسب‌ها را به‌ترتیب در جدول درج کنید:

ui, api, timeout, payment, login, bug, crash, login

بخش سوم) خروجی نهایی

پس از پایان تمام درج‌ها:

1. وضعیت نهایی جدول هش را به‌صورت زیر نمایش دهید:

0 خانه [...] :

1 خانه [...] :

...

خروجی نمونه (برای ورودی داده شده)

0 خانه: []

1 خانه: []

2 خانه: ['ui']

3 خانه: []

4 خانه: ['api']

5 خانه: ['timeout']

6 خانه: ['payment']

7 خانه: ['login', 'login']

8 خانه: ['bug']

9 خانه: ['crash']

ظرفیت نهایی جدول: ۱۰

تعداد عناصر: ۸

نکات مهم

- ترتیب عناصر داخل هر خانه اهمیت دارد.
- عملیات Rehash فقط زمانی انجام می شود که شرط ظرفیت برقرار شود.
- استفاده از dict آماده‌ی پایتون مجاز نیست (هدف، پیاده‌سازی مفهومی است).