11.1.کولیژن یا تصادم را تعریف کنید

حل : اگر برای دو داده متفاوت مقدار تابع هش یکسان دراید دو داده باید در یک خانه قرار بگیرند به این پیشامد تصادم میگویند.

11.4.2.سیستم اپن ادرسینگ در هش را توضیح دهید

حل : هر گاه برای دو یا چند داده یک خانه از تابع هش بیرون میاید تایع هش دوباره برای این داده صدا میشود یعنی اگر خانه ای مرتبط با داده مذبور پر بوده باشد تابع برای این داده با یک مقدار جدید صدا میشود تا خانه ای دیگر به ان اختصاص داده شود ، این امر نه تنها از تصادم جلوگیری میکند بلکه داده ها را در جدول پخش میکند و مانع تجمع در یک محدوده از حافظه میشود.که انواع مختلفی تابع میتوان برای ان تعریف نمود مثلا بررسی خطی و درجه دو.

11.4.3 برای جلوگیری از تصادم راه حلی ارایه دهید

حل : راه های مختلفی نظیذ گرفتن لیست پیوندی برای ذخیره موارد با ادرس هش یکسان در یک خانه و نیز استفاده از سیستم اپن ادرسینگ برای جلوگیری از تصادم میتواند مورد استقاده قرار گیرد.

.داده های زیر را در یک جدول هش به گونه ای ذخیره کنید که هیچ تداخل یا کولیژنی رخ ندهد.(میدانیم حداکثر دو داده با حروف اول یکسان وجود دارد و همگی حروف انگلیسی هستند)

AA aa Bd ss Ds hj jj mj xy xa cd

حل یک جدول هش در نظر میگیریم که حداقل خانه داشته باشد با استفاده ار سیستم ادرس دهی اپن ادرسیتگ داده ها را در جای مناسب قرار میدهیم

11.3.5 تعداد زیادی داده با مقادیر بزرگ داریم داریم ، یک تابع هش برای آن بیان کنید.

حل : یک ارایه با سایز بزرگ در نظر گرفته به طوری که تعداد داده ذخیره شده در هر سطر از O(1) باشد ، حال مقدار کد اسکی داده ها را برای هر داده (فرض کنید همه رشته هستند) تا دهمین عدد اول برای هر عدد متناسب با خانه عدد در عدد ضرب کرده همه را با هم جمع میکنیم و در نهایت به تعداد خانه خود تقسیم میکنیم ، در خانه اول هر سطح تعداد داده ذخیره شده در سطر نگه داری میکنیم و اگر تعداد از 10000 تا بیشتر شد از روش open addressing استفاده میکنیم در غیر این صورت به صورت زنجیره در یک سطح قرار میدهیم .

11.3.6. کد قسمت قبل را بنویسید .

```
Vector<int> prim(10, 2,3,5,7,11,13,17,19,23,29);

Vector<vector<string> > data; sum = 0;

String s ; Cin>> s; for(int I = 0; i < 29; I++){ if( s.size() < i){break;} else sum = prim[i] * s[prime[i]];}

Int index = sum / size; if(is_more_than_thou(data[index]) ){ h(s, 1)}  else {
data[index].push_back(s)}
```

;(index)fاین تابع به خانه اول وکتور در ردیف شماره indexیک واحد اضافه میکند .

11.7. فایده جدول درهم سازی نسبت به سایر ساختمان های داده چیست؟

حل : در واقع هش یک دیکشنری است بدین معنا که حذف و اضافه کردن و جست و جو در آن با هزنیه O(1) انجام میشود ، اما ممکن است میزان مصرف حافظه ان بیشتر باشد .

11.3.8. هدف از مکاشفه برای یک تابغ هش مناسب چیست ؟

حل : کاهش تعداد تصادم ها و نیز استفاده بهینه از حافظه موجود تا جایی که تعداد خانه های خالی به حداقل برسد . علاوه بر این میتوان جست و جو در زمان واقعی را نیز به آن افزود.

11.9. اگر اسامی تمامی خیابان های کلان شهر های تهران را بخواهیم در یک جدول هش نگهداری کنیم به گونه ای که تمامی خیابان های متقاطع را سریعا بیابیم تا در یک برنامه نقشه و راهبری استفاده کنیم چه روشی را ارایه میکنید.

حل : برای هر خیابان بعد از ذخیره در محل مورد نظر در جدول هش ایندکس تمام خیابان های متقاطع با آن را در آن خانه اضافه میکنیم و نام خیابان متقاطع را نیز در جای خود قرار میدهیم و همین کار را برای ان انجام میدهیم بدین صورت با جست و جوی یک خیابان به ترتیب نواحی اطراف ان رویت میشود .باید به صورت ریکر سیو خیابان ها نمایش داده شود.

11.10. اگر بخواهیم تمام اطلاعات ایرانیان را در یک جدول درهم ذخیره کنیم ، با توجه به این که شماره ملی افراد یکتاست ولی از صفر و یا یک شروع نمیشود چه راهی را دنبال میکنیم؟

حل : ابتدا با توجه به سه رقم اول شماره ملی خانه های جدول را شماره گذاری میکنیم و سپس برای هر 3 رقم مجزا شده 100000 خانه اختصاص میدهیم ، حال برای هر شماره ابتدا 3 رقم را تفکیک میکنیم و بعد باقی مانده 7 رقم باقی را به 100000 میگیریم اگر خانه پر بود از اپن ادرسینگ استفاده میکنیم ، اما روش کار به این صورت است که ابتدا بدون اپن ادرسینگ انجام میدهیم بعد از آن وقتی همه به صورت طبیعی در جای خود قرار گرفتند از این روش بهره میگیریم اگر فردی اضافه شد او را به خانه دقیق خود میبریم و برای فردی که میخواهیم جابه جا کنیم اپن ادرس میکنیم .