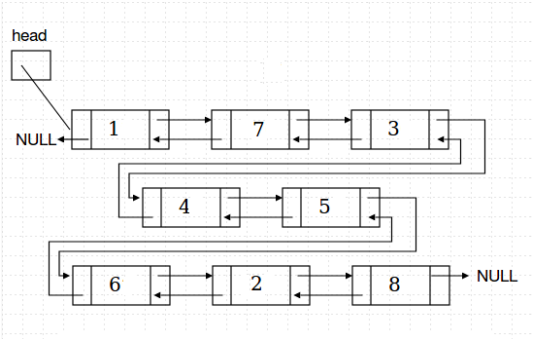
تمرین سری دوم

درس ساختمان داده، پاییز ۱۴۰۲
دانشگاه صنعتی خواجه نصیرالدین طوسی ـ دانشکده ریاضی

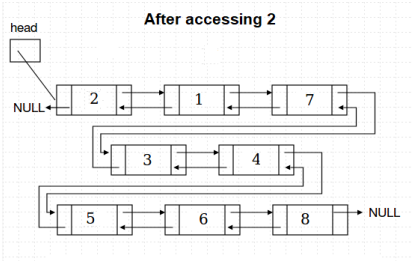
(پیاده سازی) فرض کنید یک لیست پیوندی دوسویه داریم که حاوی جایگشتی از اعداد ۱ تا n است. اینها در واقع کلید عناصر داخل لیست هستند. برای مثال یک لیست برای $n = 8$ می‌تواند بصورت زیر باشد.

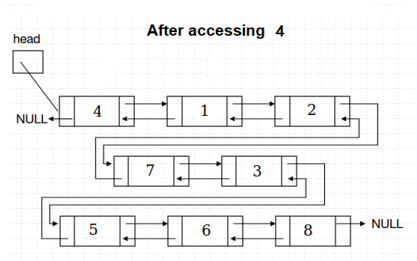


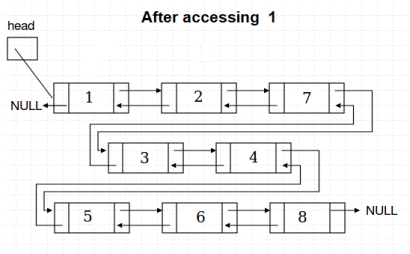
فرض کنید کلیدهای داخل لیست به کرات مورد دسترسی قرار می‌گیرند. در دسترسی به کلید $key = i$ لیست از اول پیمایش شده تا به کلید i برسیم. پس هزینه دسترسی برای کلید i فاصله i از اول لیست است. برای مثال در شکل بالا هزینه دسترسی به کلید $key = 2$ برابر با 6 است.
هزینه یک دنباله از کلیدها، مجموع هزینه های دسترسی به این کلیدهاست. برای مثال در شکل بالا،

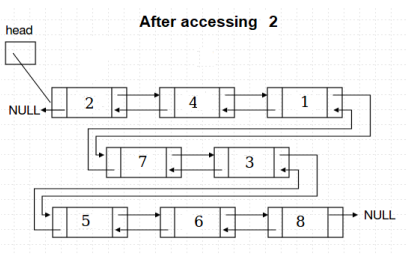
$$Cost(2, 1, 4, 2) = Cost(2) + Cost(1) + Cost(4) + Cost(2) = 6 + 0 + 3 + 6 = 15$$

یک ترفند برای کاهش هزینه دسترسی در لیست پیوندی، آوردن کلید مورد دسترسی قرار گرفته به اول لیست است. یعنی هر بار کلید i مورد دسترسی قرار گرفت آن را به اول لیست می آوریم. پس اینجا ساختار لیست پیوندی بعد از هر دسترسی ممکن است تغییر کند. هزینه آوردن یک عنصر به اول لیست را شما عدد ثابت c در نظر بگیرید. با این فرض با داشتن دنباله دسترسی $2, 1, 4, 2$ لیست پیوندی شکل بالا بصورت زیر تغییر می‌کند.









با این ترفند هزینه دسترسی به دنباله $2, 1, 4, 2$ بصورت زیر در می آید.

$$Cost'(2,1,4,2) = Cost'(2)+Cost'(1)+Cost'(4)+Cost'(2) = (6+c)+(1+c)+(4+c)+(2+c) = 13+4c$$

در این تمرین می‌خواهیم لیست پیوندی دو سویه را پیاده‌سازی کنیم و این دو استراتژی (حالتی که لیست تغییر نمی‌کند و حالتی که لیست تغییر می‌کند) را با هم مقایسه کنیم. ساختار داده ای که ایجاد می‌کنید باید عمل $access(i)$ را برای هر دو استراتژی پیاده‌سازی کند.
فرض کنید $n = 10000$ و در شروع کار کلیدها در لیست پیوندی با یک ترتیب تصادفی ذخیره شده‌اند. زمان اجرا را برای دو استراتژی در حالتهای زیر مقایسه کنید.

۱. دنباله دسترسی $1, 2, 3, \cdots, n-1, n$

۲. دنباله دسترسی $\underbrace{1, \cdots, 1}_{1000}, \underbrace{2, \cdots, 2}_{1000}, \cdots, \underbrace{10, \cdots, 10}_{1000}$

۳. وقتیکه دنباله دسترسی توسط کد زیر تولید شود.

```
import numpy as np
np.random.seed(0)
a = np.random.normal(5000, 1000, size=10000)
a = a.round(decimals=0, out=None)
a = abs(a)
a = a[a < 10000 ]
```

کد همراه با نتیجه محاسبات در قالب یک فایل پی دی اف تحویل داده شود.