پروژه سوم آخرین مهلت تحویل: ۱۴۰۱/۱۰/۳۰

پیاده‌سازی الگوریتم حریصانه‌ی Prim

هدف از این پروژه پیاده‌سازی الگوریتم Prim برای یافتن درخت پوشای می‌نیمم (Minimum-Weighted-Spanning-Tree) است.

ابتدا فایل‌های زیر را که حاوی کلاس‌های جاوا هستند، دانلود کنید.

۱. **Graph.java** : این فایل حاوی سه کلاس است: ۱- کلاس **Graph** که گراف را تعریف می‌کند، ۲- کلاس **Node** که راس‌های گراف را تعریف می‌کند، و ۳- کلاس **Edge** که لبه‌های گراف را تعریف می‌کند.
۲. **Prim.java** : کلاسی است که الگوریتم **Prim** را تعریف و پیاده‌سازی می‌کند.
۳. **Test_MST_Prim.java** : می‌توانید از این کلاس برای تست برنامه نهایی خود استفاده کنید. توصیه می‌شود تست را بر روی گراف‌های دیگری نیز به جز آنچه در این کلاس ارائه شده انجام دهید.

کد موجود در **Graph.java** و کد موجود در **Prim.java** را با نوشتن کدی که هر یک از رویه‌های زیر را پیاده‌سازی می‌کند کامل کنید. (تعریف رویه‌ها شامل نام و پارامترها را به هیچ وجه تغییر ندهید):

۱. public void **addEdge**(String sourceLabel, String destinationLabel, int weight)
در کلاس **Graph.java**
این رویه یک لبه به گراف اضافه می‌کند. توجه کنید که وقتی **Test_MST_Prim** اجرا می‌شود در**main** ابتدا برچسب راس‌های گراف اختصاص داده می‌شود. سپس لبه‌ها با فراخوانی این رویه به گراف اضافه می‌شوند.
دقت کنید چون گراف غیرجهت‌دار (undirected) است، هر لبه‌ای مانند a – b باید از هر دو راس قابل دسترسی باشد. بنابراین هر لبه باید به مجموعه لبه‌های هر دو راس a و b اضافه شود.

۲. public void **solveMST**(Graph g, String root)
در کلاس **Prim.java**
این رویه الگوریتم **Prim** را پیاده‌سازی می‌کند و درخت پوشای می‌نیمم را در متغیر کلاس **ArrayList<Node> msTree** و هزینه یا مجموع وزن‌های لبه‌های درخت را در متغیر کلاس **int mstCost** قرار می‌دهد. از شبه کد ارائه شده در درس مربوطه برای پیاده‌سازی الگوریتم استفاده کنید. (از کلاس PriorityQueue<...> خود جاوا برای پیاده‌سازی صف اولویت استفاده کنید.)

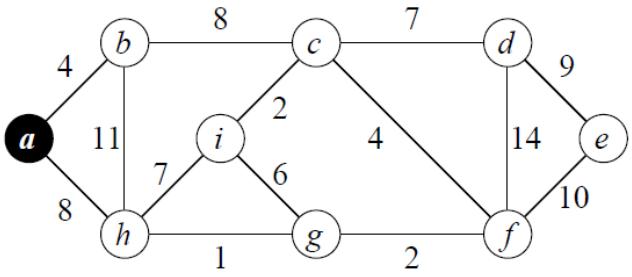
توجه: به هیچ وجه قسمت‌های دیگر فایل‌ها را تغییر ندهید.

آنچه باید تحویل دهید:

تنها دو فایل **Graph.java** و **Prim.java** را آپلود کنید.
– نام خود را در ابتدای دو فایل به صورت یک کامنت (...//) قرار دهید.
– برنامه را حتما در جاوا بنویسید.
– برای پیاده‌سازی اجازه استفاده از هیچ کتابخانه‌ای به جز آنچه در متن فایل‌های پروژه ذکر شده است، ندارید.
– رویه‌ها را درست به صورتی که در این متن توضیح داده شده است، پیاده‌سازی کنید.

یک نمونه خروجی برنامه:

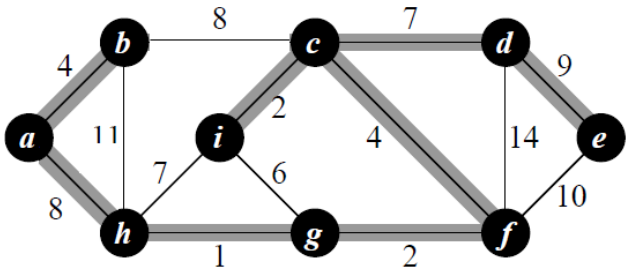
در فایل **Test_MST_Prim.java** ابتدا گراف زیر که در کتاب مرجع درسی CLRS (صفحه ۶۳۵) نیز موجود است، ایجاد می‌شود.



سپس درخت پوشای می‌نیمم با شروع از راس a استخراج می‌گردد. با پیاده‌سازی درست دو رویه‌ی ذکر شده در بالا، خروجی برنامه به صورت زیر خواهد بود.

```
Root a
Edge a -- b Weight= 4
Edge a -- h Weight= 8
Edge h -- g Weight= 1
Edge g -- f Weight= 2
Edge f -- c Weight= 4
Edge c -- i Weight= 2
Edge c -- d Weight= 7
Edge d -- e Weight= 9
Total MST cost: 37
```

که معادل درخت زیر است.



توجه کنید که این گراف بیش از یک درخت پوشای می‌نیمم دارد. (درخت پوشای موجود در کتاب درخت متفاوتی است)