



به نام خدا  
سیستم‌های توزیع شده  
۱۳۹۸-۲

تمرین دوم  
مدرس: صابر صالح

### نکات مهم

لطفا ابتدا به نکات زیر توجه کنید:

- برای پیاده‌سازی این تمرین از زبان Python استفاده نمایید.

- مهلت ارسال تمرین تا پایان روز ۱۳۹۹/۰۲/۰۷ می‌باشد.

موفق باشید.

## ۱ مقدمه

در برخی کاربردها در سیستم های توزیع شده، نودها اطلاعی از سائز شبکه ندارند. اگر نودی بخواهد از سائز شبکه (تعداد نودها) باخبر شود الگوریتم های مختلفی برای این کار وجود دارد که در این تمرین می خواهیم به بررسی و پیاده سازی دو مورد از آن ها پردازیم.

به هر یال در این تمرین یک تاخیر ارسال نسبت داده می شود.

توجه شود که گراف شبکه بدون جهت است.

برای ارسال پیام بین گره ها از *socket programming* استفاده کنید.

## ۲ الگوریتمی بر پایه Flooding

به طور کلی این الگوریتم ابتدا یک درخت فراگیر می یابد و سپس با استفاده از این درخت فراگیر، پیام هایی منتقل می شود که در نهایت ریشه ی این درخت از سائز شبکه باخبر می شود.

## ۱.۲ توضیح مرحله به مرحله الگوریتم

۱. در ابتدا نودی که می خواهد از سائز شبکه با خبر شود پیام search را برای همسایه های خود می فرستد.

۲. هر نود که پیام search را دریافت کرد روند زیر را طی می کند:

□ اگر نود مذکور قبلا mark نشده باشد:

(آ) خود را mark می کند.

(ب) نود فرستنده پیام search را به عنوان parent خود تعیین می کند.

(ج) پیام parent را برای فرستنده ی پیام search می فرستد.

(د) به همه همسایه هایش پیام search را می فرستد.

□ اگر نود مذکور قبلاً mark شده باشد نود پیام nonparent را برای فرستنده پیام search می فرستد.

۳. هر نود که پیام parent را دریافت کند نود فرستنده پیام را به مجموعه ی children خود اضافه می کند.

۴. اگر نودی از همه ی همسایه هایش پیام nonparent دریافت کرد متوجه می شود که برگ درخت فراگیر است و الگوریتم convergecast را آغاز می کند.

۵. هر نود پیام convergecast را برای parent خود می فرستد. به این فکر کنید که این پیام حاوی چه اطلاعاتی باشد و چه زمانی ارسال شود که در نهایت root بتواند سائز شبکه را بفهمد.

## ۲.۲ ورودی‌ها

برای ساختن گراف شبکه یالها در ورودی داده می‌شوند. هر یال به کمک سه تایی شناسه‌ی گره اول، شناسه‌ی گره دوم و تاخیر ارسال مشخص می‌گردد.

نخستین عددی که وارد می‌شود، شناسه‌ی گرهی است که می‌خواهد سائز شبکه را بداند.

node1	node2	delay
0	1	2
0	2	1
1	2	3
1	3	1
1	4	2
3	2	4
4	3	2

پایان ورودی‌ها با یک "\n" اضافه تعیین می‌شود. نمونه‌ی ورودی:

---

## ۳.۲ خروجی ها

در خروجی باید log پیام های ارسالی توسط نودها چاپ شود.

(از آن جایی که این تمرین تحویل حضوری دارد فرمت خاصی برای چاپ این log مدنظر نیست ولی خروجی باید واضح و کامل باشد.)

در نهایت و در سطر آخر تعداد نودها از نظر نود ریشه چاپ شود.

### ۳ الگوریتمی بر پایه Gossip (امتیازی)

به طور کلی الگوریتم های *gossip* برای با خبر شدن از اطلاعاتی در کل سطح گراف است اما نه به طور قطعی و دقیق بلکه به صورت تقریبی. در این تمرین می خواهیم با به کار گیری الگوریتمی بر پایه *gossip* هر کدام از گره های شبکه بتوانند سائز شبکه را تخمین بزنند.

روند الگوریتم های بر پایه *gossip* به این شکل است که هر گره در شبکه به کمک تکرار فرآیندی ثابت با همسایه های خود می خواهد، اطلاعاتی از کل گراف بدست آورد.

فرض کنید هر گره دارای برداری شامل ۱۰۰ عدد است که این اعداد دارای توزیع نمایی با پارامتر ۱ هستند و به صورت مستقل تولید شده اند. می خواهیم به کمک بردار گفته شده، هر گره بتواند سائز شبکه را تخمین بزند. خاصیت سودمند زیر را در فرآیند حل به خاطر داشته باشید:

$$X \sim \exp(\alpha), Y \sim \exp(\beta) \Rightarrow \min(X, Y) \sim \exp(\alpha + \beta)$$

#### ۱.۳ توضیح مرحله به مرحله الگوریتم

۱. هر گره در شبکه برداری به طول ۱۰۰ از اعداد تصادفی نمایی با پارامتر یک برای خود می سازد.
۲. هر ۲۰۰ میلی ثانیه یکی از همسایه های خود را به تصادف انتخاب کرده و بردار خود را برای او می فرستد.
۳. هر گره ای که برداری دریافت کرد بین بردار خود و بردار دریافتی عملیاتی انجام می دهد. (تشخیص عملیات مفید برای اینکه در نهایت بتوانید به کمک این بردار سائز گره های شبکه را تخمین بزنید با شماست).
۴. این فرآیند به مدت یک دقیقه ادامه پیدا میکند.
۵. بعد از یک دقیقه هر کدام از گره ها به کمک بردار خود و روشی که شما پیاده سازی میکنید سائز شبکه را تخمین میزند.

## ۲.۳ ورودی ها

برای ساختن گراف شبکه یالها در ورودی داده می شوند. هر یال به کمک سه تایی **شناسه ی گره اول، شناسه ی گره دوم و تاخیر ارسال** مشخص می گردد.

نخستین عددی که وارد می شود، شناسه ی گرهی است که می خواهد سائز شبکه را بداند.

```
node1 node2 delay
```

پایان ورودی ها با یک "\n" اضافه تعیین می شود.

برای اینکه الگوریتم در زمان یک دقیقه همگرا شود تاخیر ها بین ۱۰ تا ۱۰۰ میلی ثانیه خواهند بود و واحد آن ها در ورودی میلی ثانیه خواهد بود.

نمونه ی ورودی: ۱ ۱ ۱

```
0 1 20
0 2 100
1 2 30
1 3 10
1 4 20
3 2 40
```

## ۳.۳ خروجی ها

خروجی ها به این صورت است که شامل  $k_i$  خط مجزا است و  $k_i$  تعداد شناسه های گره ها می باشد و در هر خط ابتدا شناسه ی گره و سپس تخمین سائز شبکه توسط آن گره نوشته میشود و خروجی به شکل زیر است.

```
nodeID1 EstimatedNetworkSize1
nodeID2 EstimatedNetworkSize2
nodeID3 EstimatedNetworkSize3
. .
. .
. .
```