گزارش پروژه دوم
نگار میرگتی ۸۱۰۱۹۴۴۱۳

سوال اول:
بخش الف) هدف کمینه کردن عبارت مقابل است :

$\sum P(v)$, $v \notin V(T)$ + $\sum c(e)$, $e \in E(T)$

که در آن p(v) مقدار prize های تعیین شده برای راس ها است و c(e) هزینه یال ها است.

الگوریتم Heuristic پیشنهادی که از الگوریتم 2-approximation مطرح شده در کلاس برای پیدا کردن درخت اشتاینر مینیمم استفاده می کند :

```
S = Terminal Nodes in G
prev_cost = INF
new_cost = ∑weight(e) , E ∈ E(G)

While(new_cost <=prev_cost)
   Prev_cost = new_cost
   Construct Gprime = (Vprime, Eprime) where Vprime = S and edges in Eprime correspond
   To shortest paths between vertices in G
   Find MST T of Gprime  , new_cost = ∑cost(e), e∈E(T)
   Convert T to original graph G and obtain a subgraph Tprime , S = Vertices of Tprime
Endwhile

#Remove unnecessary leaves
for all leaves v of Tprime
   if Prize(v) < Connection cost(v)
      remove v from Tprime
endfor
```

در هر مرحله حداقل یک راس اشتاینر به گراف اضافه می شود(درغیر اینصورت از حلقه خارج می شویم چرا که الگوریتم نتوانسته به وزنی کمتر از وزن مرحله قبلی برسد) در نتیجه حلقه ی while حداکثر v بار تکرار می شود.
اگرهزینه الگوریتم پریم را O(Elogv) در نظر بگیریم هزینه نهایی برابر خواهد بود با:

O(E^2Logv + V^2)

سوال دوم :
بخش الف) جدول خروجی در فایل output2.html قرار داده شده است. برای بدست آوردن جواب گراف بدون جهت از directed=FALSE در آرگومان توابع استفاده شده است. همچنین برای حالاتی که این آرگومان وجود نداشت گراف بدون جهت (g_net_undir <- as.undirected(g_net, mode="collapse" به عنوان ورودی به تابع داده شده است.

بخش پ) خروجی برای توابع شباهت igraph:

```
dice : Pycelle
jaccard : Pycelle
invlogweighted : Eddard Stark
```

بخش ت)
روابط به کار رفته در محاسبه :

$$M(i,j) = \frac{a_{ij}}{\sum_{j=1}^{n} a_{ij}}$$

$$C_i^{RWC} = \frac{n}{\sum_{j=1}^{n} H(j,i)}$$

For computational convenience, this expression can be vectorized as

$$H(.,j) = (I - M_{-j})^{-1} e$$

where $H(.,j)$ is the vector for first passage times for a walk ending at node j, and e is an n-1 dimensional vector of ones.

Mean first passage time is not symmetric, even for undirected graphs.

خروجی الگوریتم Random Walk Closeness Centrality :

42 39 68 7 64 2 242 96 240 85

خروجی تابع بخش الف :

23, 4, 240, 44, 64, 52, 156, 68, 56, 42

۴ راس بین دو خروجی مشترک است.