Approximation Algorithms
Problem set #4

The assignment is due Wednesday, May 31. Please, submit your work via Canvas.

Problem 1 Design dynamic programming algorithms for finding the exact (a) Minimum Balanced Cut, (b) Minimum Multiway Cut on edge weighted trees.

(c) Prove that in every tree there exists a sparsest cut that cuts exactly one edge. Design an algorithm for finding the exact sparsest cut on trees.

Problem 2 Design a poly-logarithmic approximation algorithm for the Minimum Linear Arrangement Problem. In this problem, we are given a graph $G = (V, E)$ on n vertices, and the goal is to find a one-to-one mapping $f: V \to \{1, \dots, n\}$ to minimize the total length of all edges: $\sum_{(u,v)\in E} |f(u) - f(v)|$.

a. Show that for any solution the objective (defined above) equals to
$$\sum_{i=1}^{n-1} |\{(u, v) \in E: f(u) \leq i \text{ and } f(v) > i\}|$$
b. Consider the optimal solution f^* with cost OPT. Let $A_i^* = \{u: f^*(u) \leq i\}$; $B_i^* = \{v: f^*(v) > i\}$. Show that for some $i \in \left[\frac{n}{4}, \frac{3n}{4}\right]$, the size of the cut (A_i^*, B_i^*) is at most $O(\mathrm{OPT}/n)$. That is, there exists an *almost* balanced cut in S of cost $O(\mathrm{OPT}/n)$.

For parts c and d, assume that you have a $O(\log n)$ approximation algorithm for finding the minimum cut (A, B) with $A = k$, $B = n - k$. (Optional question: How would you design such an algorithm using embeddings into Dominating Trees?)

c. Consider a subset $S \subset V$. Let
$$OPT(S) = \sum_{\substack{(u,v)\in E \\ u,v\in S}} |f^*(u) - f^*(v)|$$
Show that there exists an almost balanced cut in S of cost at most $O(\mathrm{OPT}(S)/|S|)$.

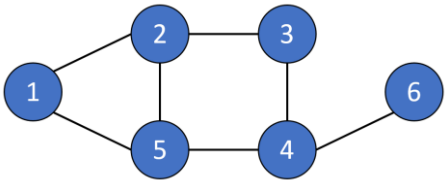
d. Using parts a, b, and c. Design a recursive approximation algorithm for finding the minimum linear arrangement.
e. What is the depth of the recursion? How many edges does the algorithm cut at every level of the recursion and what is their cost (total length)?

Laplacian – Definition and Examples Consider an (undirected) graph $G = (V, E)$ on n vertices. The Laplacian of G is the quadratic form $L(x)$ defined as follows:
$$L(x) = \sum_{(u,v)\in E} |x_u - x_v|^2,$$

where x is an n dimensional vector indexed by the vertices of the graph. The corresponding matrix L equals

$$L_{uv} = \begin{cases} \deg(v), \text{if } u = v \\ -1, \text{if } (u, v) \in E \\ 0, \text{otherwise} \end{cases}$$

Here is an example:



Graph

$$\begin{pmatrix} 2 & -1 & 0 & 0 & -1 & 0 \\ -1 & 3 & -1 & 0 & -1 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & -1 \\ -1 & -1 & 0 & -1 & 3 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 \end{pmatrix}$$

Laplacian

There are many tools for finding the eigenvalues and eigenvectors of a matrix/linear operator (e.g., MATLAB and R). Here is how it can be done with R.

```
> L = matrix(c(2, -1, 0, 0, -1, 0,-1, 3,-1,0,-1, 0, 0, -1, 2, -1,
               0, 0, 0, 0, -1, 3, -1, -1, -1, -1, 0, -1, 3, 0, 0,
               0, 0, -1, 0, 1), 6, 6)
> x = eigen (L)
> x
```

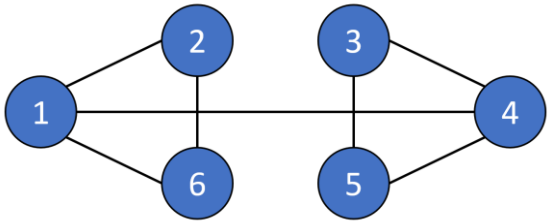
R Program

```
$values

[1] 4.891220e+00 3.704624e+00 3.000000e+00 1.682569e+00 7.215864e-01
[6] 3.552714e-15

$vectors

      [,1]    [,2]    [,3]    [,4]    [,5]   [,6]

[1,]  0.0323   0.5670  -0.2886   0.5052  -0.4148  0.4082
[2,]  0.4685  -0.6581  -0.2886   0.0402  -0.3094  0.4082
[3,] -0.3564   0.2051  -0.2886  -0.7590  -0.0692  0.4083
[4,]  0.5619   0.3084   0.5773  -0.2006   0.2209  0.4082
[5,] -0.5619  -0.3084   0.5773   0.2006  -0.2209  0.4082
[6,] -0.1444  -0.1140  -0.2886   0.2939   0.7935  0.4082
```

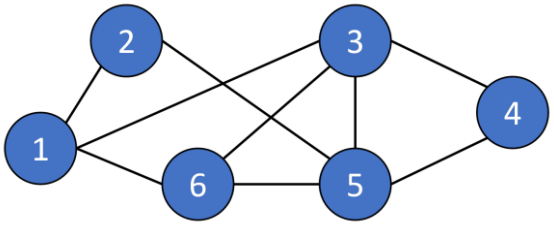
R Output

The second eigenvector is in the 5th column of the matrix: (-0.4148, -0.4148, -0.0692, 0.2209, -0.2209, 0.7935).

Problem 3 Write Laplacian matrices for the following graphs. Compute the second eigenvector for each of them (you can use MATLAB, R, or any other software). Find all threshold cuts i.e. cuts of the form $A = \{u: x_u \leq t\}$ and $B = \{v: x_v > t\}$.



a



b