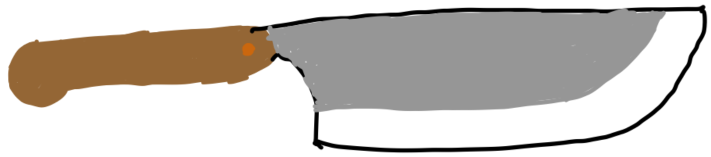
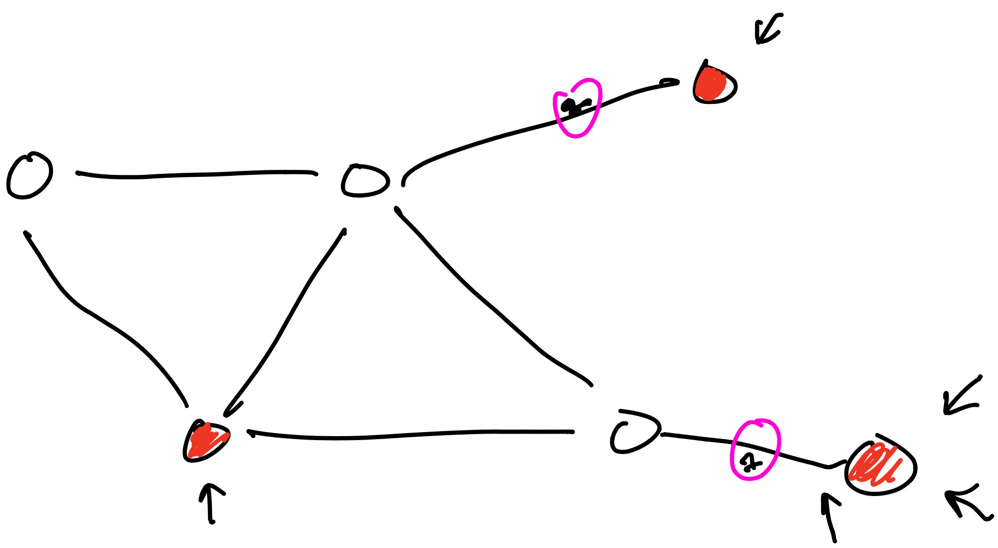
Approximation Algorithms for Cut Problems



Ch. 4 of Vazirani's Approximation Algorithms

<u>Multiway Cut Problem</u>: Given an undirected, connected, weighted $G = (V, E)$. Given a set $S = \{s_1, s_2, \ldots, s_k\}$. A <u>multiway-cut</u> is a set of edges whose removal disconnects all the s_i's from each other.

<u>min-weight</u>



<u>Min k-cut problem</u>: A k-cut is a set of

edges whose removal leaves k-connected components. We want min-weight such k-cut.

Multiway cut for fixed cut $k \geq 3$ is NP-Hard.

min k-cut is NP-Hard if k is part of the input.

Good news $\rightarrow$ Simple approx. algorithms.

Approx. factor $\quad 2 - \frac{2}{k}$

Multiway Cut:

An isolating cut for s_i is a min-weight cut that disconnects s_i from the other terminals.

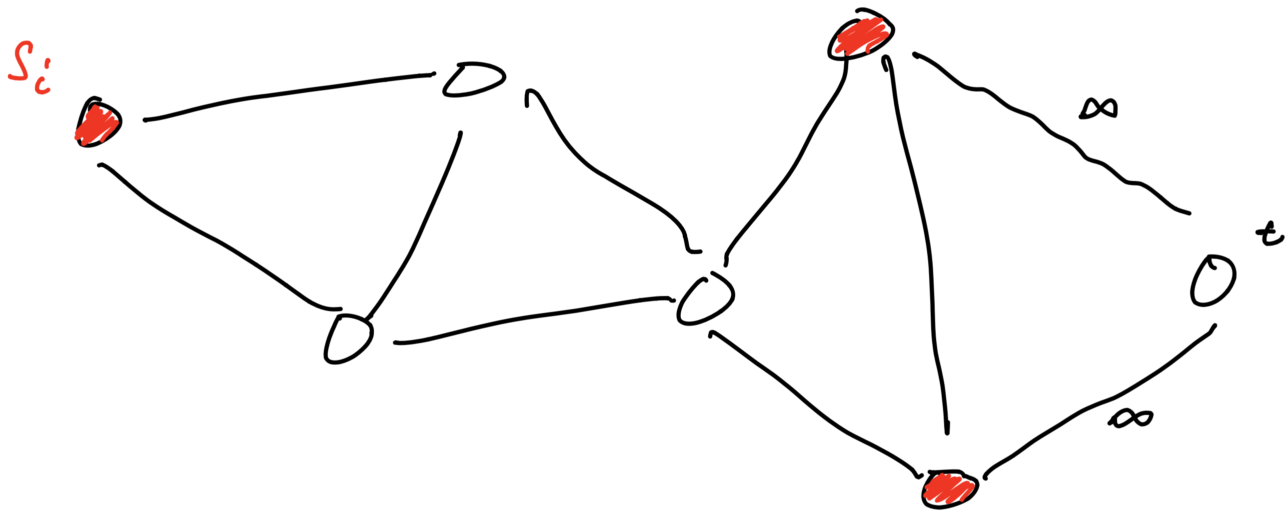
Algorithm:

1. For each s_i compute the min-weight isolating cut.

2. Discard the heaviest, output the union

of the rest [call that union c].

Step 1:



Find min $s_i - t$ cut.

Thm: MC Alg. has an approximation guarantee of $2 - \frac{2}{k}$.

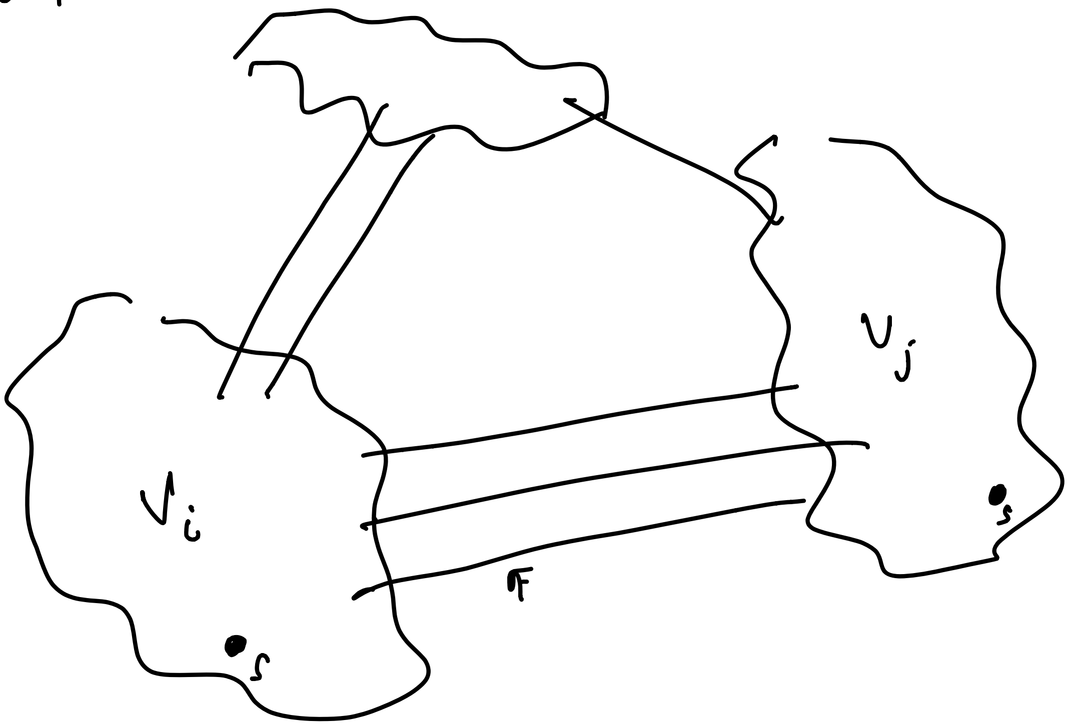
$(1 - \frac{1}{k})$
↑
discard

Proof: Let A be the optimal multiway cut in G. When we remove $A \to k$ connected components. $V_1, V_2, \ldots, V_k$. Let's call A_i the cut that disconnects V_i from the graph.

Each edge in A appears in two of the A_i's.

in A_i, there's some edge disconnecting V_i from some other V_j.

$$\sum_{i=1}^{k} w(A_i) = 2 \cdot w(A).$$



C_i is an min-weight isolating cut for S_i. we know,

$$w(C_i) \le w(A_i).$$

$$C = \bigcup_{i=1}^{k} C_i \setminus \text{"the heaviest"}$$

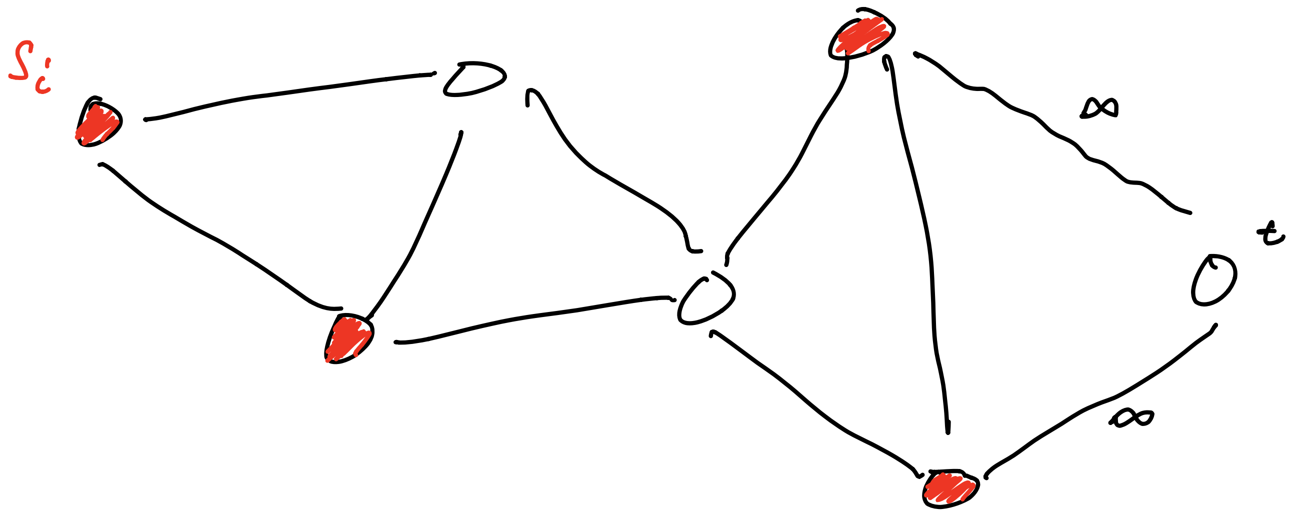
$$w(C) \le \left(1-\frac{1}{k}\right) \sum_{i=1}^{k} w(C_i) \le \left(1-\frac{1}{k}\right) \sum_{i=1}^{k} w(A_i)$$

$$\leq 2 \cdot \left(1 - \frac{1}{k}\right) w(A).$$

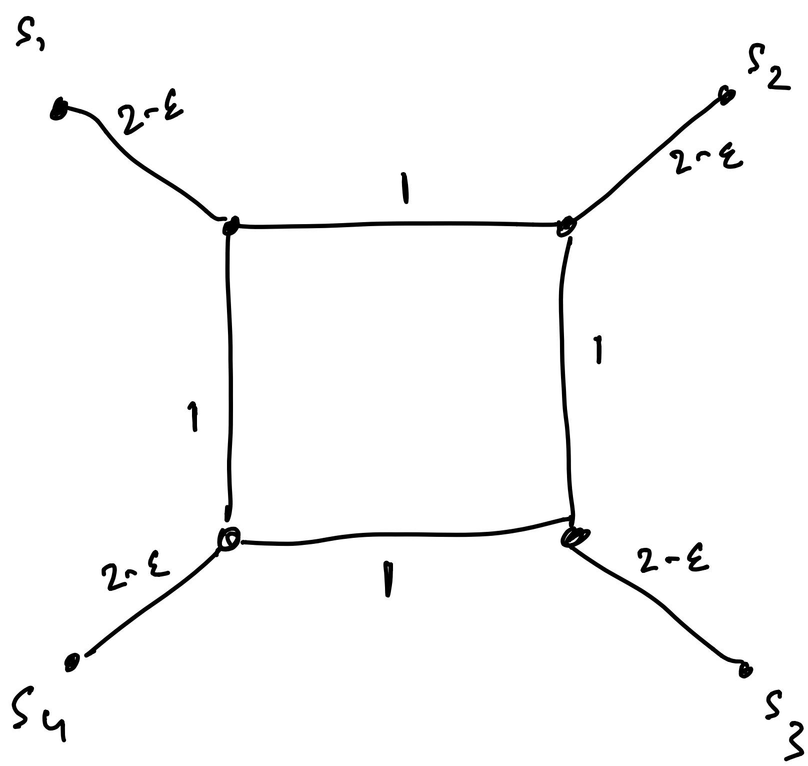
$\square$

$k = 4. \quad S = \{s_1, s_2, s_3, s_4\}$

(we can get 4/3 somehow)



Tight Example:



C

$(k-1)(2-\varepsilon)$

OPT

k

Min k-cut: Gomory-Hu trees

Let tree T be a tree on the vertex set V of G.

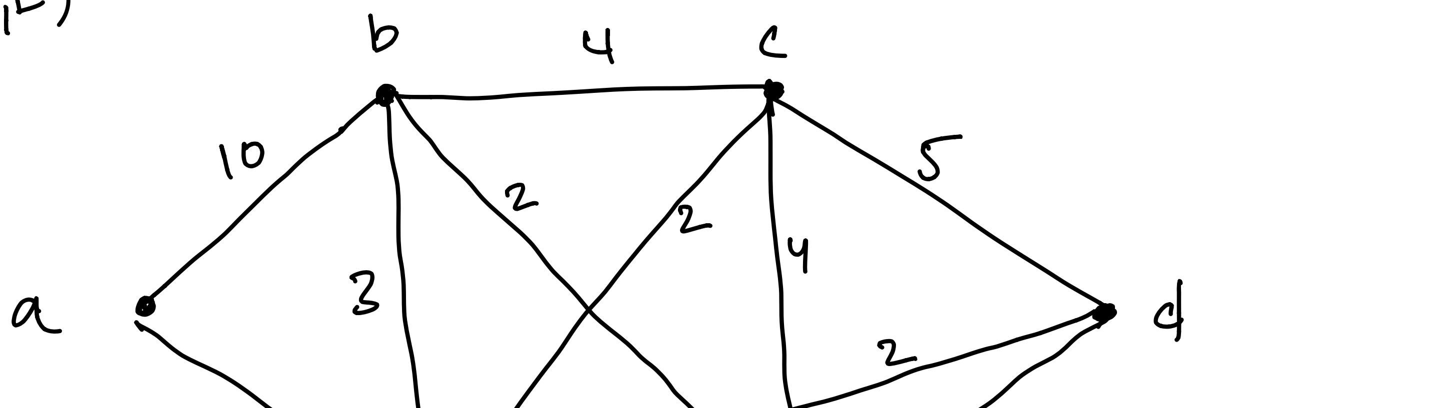
Let e be an edge in T. If we remove e, we split T into two sets of vertices $(S, \bar{S})$.

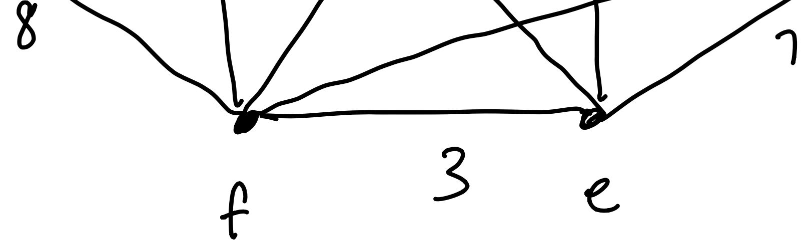
The edge e is in T is associated with the cut $(S, \bar{S})$ in G.

1. $\forall\, u, v \in T$, the weight of a min u-v cut in G is the same as the weight of a min u-v cut in T.

2. The weight of edge e in T is the weight of the cut $(S, \bar{S})$ in G.

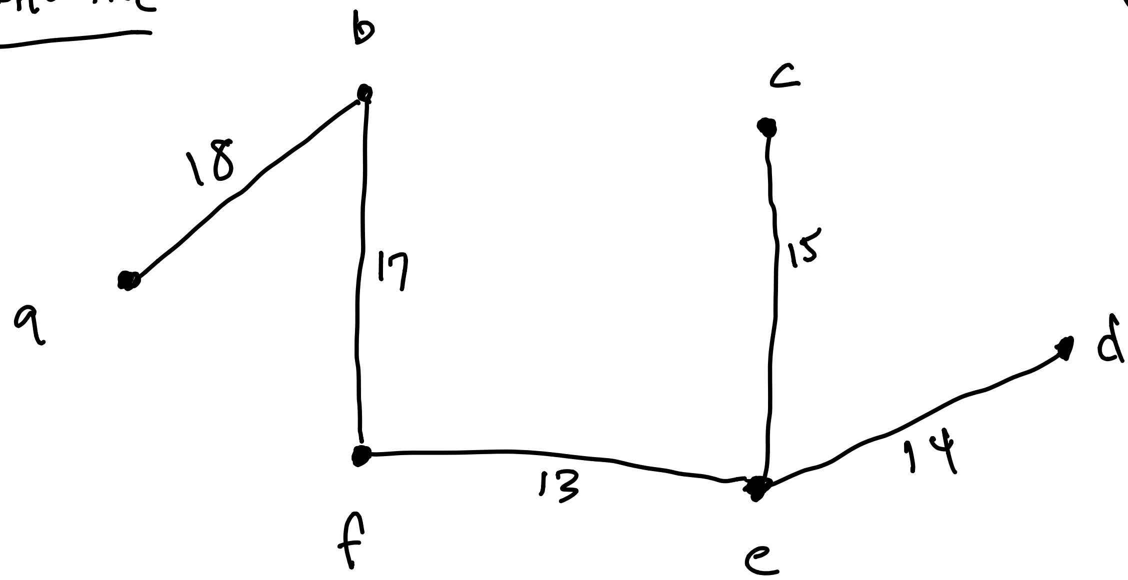
$G = (V, E)$





Gomory-Hu Tree

($|V|-1$)
max-flow
computations



Lemma: Let S be the union of cuts in G associated with l edges in T. Removing S from G leaves $l+1$ connected components behind.

Proof: Removing the l edges in T leaves $l+1$ connected components in T. Call these vertex sets $V_1, V_2, \ldots, V_{l+1}$. Removing S

from G disconnects every pair V_i, V_j.

So, we have at least $l+1$ connected components.

Alg. k-cut:

1. Compute a Gomory-Hu Tree of G
2. Output the union of the lightest $k-1$ cuts of the $n-1$ cuts in T. [call this union C].

Theorem: Alg. k-cut guarantees an approximation factor of $2 - \frac{2}{k}$.

Proof: Let A be an optimal k-cut in G. We view $A = A_1 \cup A_2 \cup \dots \cup A_k$. Let A_i be the cut that separates V_i from the rest of the graph.

Since each edse edge of A is in two of the A_i's.

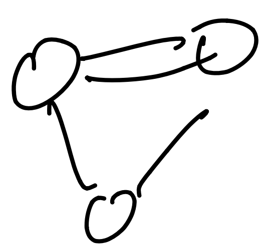
$$\sum_{i=1}^{k} w(A_i) = 2w(A)$$

(v_k)

Assume, A_k is the heaviest of these cuts.

Let B be the set of edges in T that connect across two of the sets $V_1, V_2, \ldots, V_k$.

Consider a new graph on the vertex set V and edge set B. Shrink each $V_1, V_2, \ldots, V_k$ into just one vertex. $\rightarrow$ Connected graph.



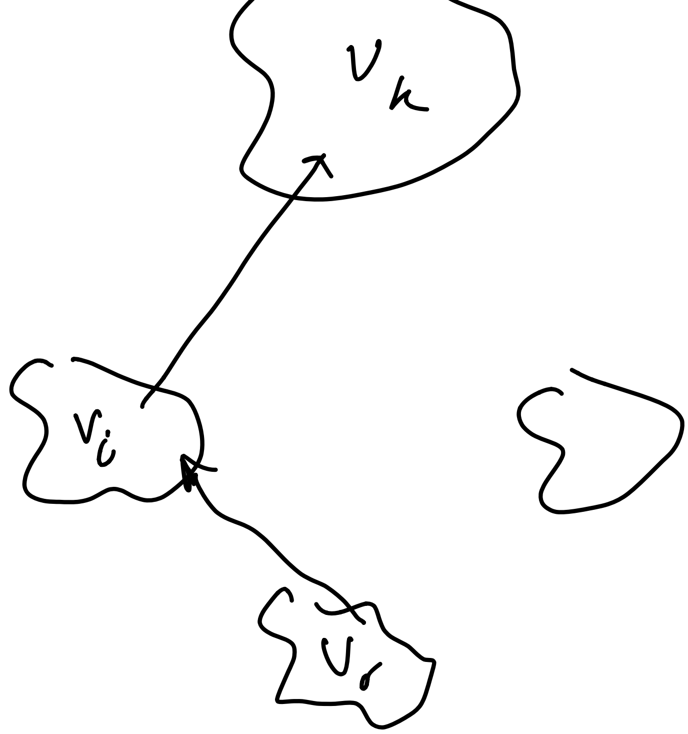
Throw away edges until we have a tree.

$B' \subseteq B$ are the leftover edges.

$|B'| = k-1$.

An edge e is associated

with the vertex set it comes out of.



$e = (u, v) \in B'$

The weight of a min. u-v cut in G is $w'(u,v)$.

A_i is a u-v cut.

$\Rightarrow$ $\underline{w'(u,v) \leq w(A_i)}$

C is the union of the lightest k-1 cuts:

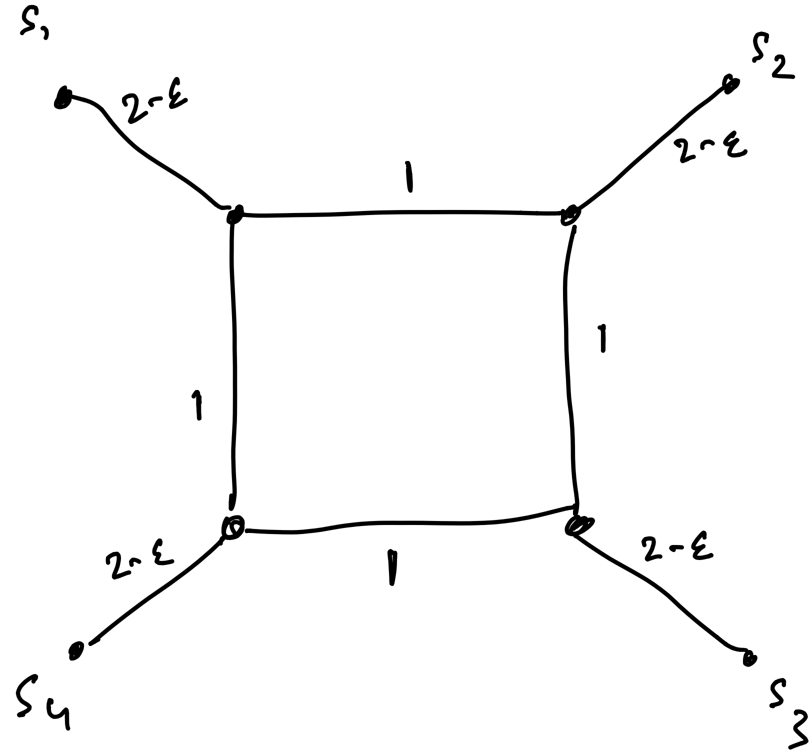
$$w(C) \leq \sum_{e \in B'} w'(e) \leq \sum_{i=1}^{k-1} w(A_i) \leq \underbrace{\left(1 - \frac{1}{k}\right) \sum_{i=1}^{k} w(A_i)}$$

$$\leq 2\left(1 - \frac{1}{k}\right) w(A).$$

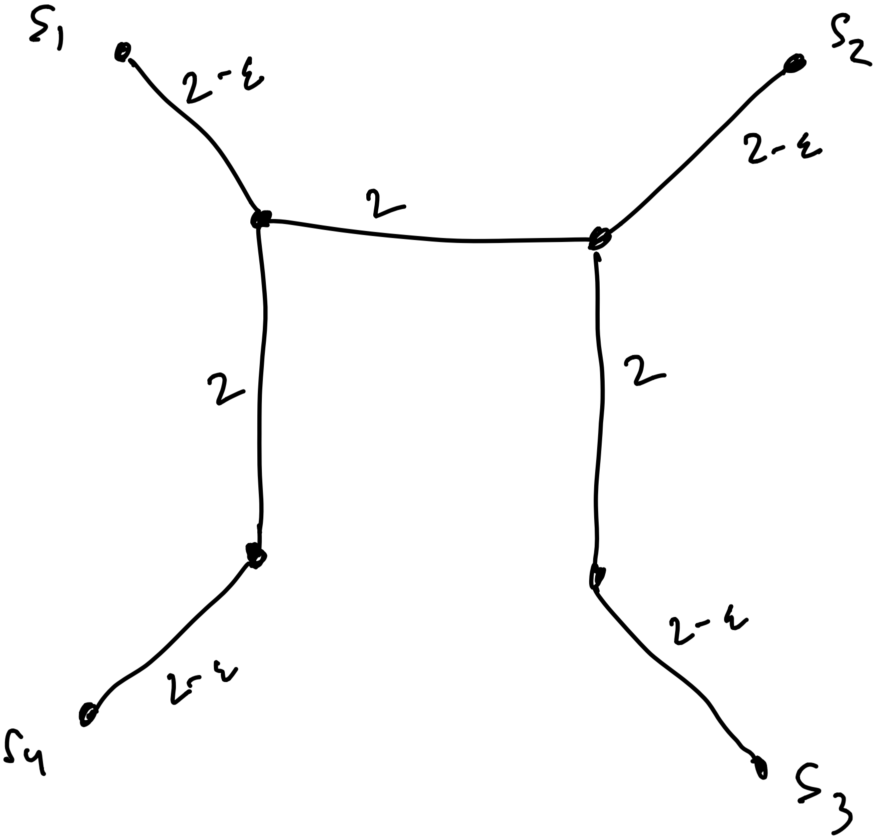
$\square$

<u>Tight Example:</u>

G:



GH tree:



C: $(k-1)(2-\varepsilon)$

$$\frac{\text{OPT } k\text{-cut}}{k}$$

Vazirani
+
Saran

Proof by R. Ravi @CMU

Thanks!! :)