Task 3

Let $G = (V, E)$ be an undirected graph with $w_{uv} > 0 \; \forall \; (u,v)$ in E.

__Def__ a 'cut(A, B)' satisfies

$$A \subseteq V, \; B = V \setminus A.$$

$$\text{cut}(A,B) = \sum_{\substack{(u,v)\in E \\ u\in A, v\in B}} w_{uv}$$

a) SEARCHMAXIMUMCUT

Let (A, B) be a partition of vertices V of graph G

while $\exists \, u \in A$ s.t. $\Delta u > 0$

move u to B

return cut(A, B)

where

$$\Delta u := \sum_{v\in B} w_{uv} - \sum_{v\in A} w_{uv},$$

Since $\Delta u > 0$ and not ≥ 0, the algorithm terminates. Also since we are always moving an element, assuming a finite graph, we must be able to run out of vertices, ending the search.

b) Assume SEARCHMAXIMUMCUT $\to$ cut(A, B) and the global optimum is cut(A^*, B^*)

__Goal__ $\text{cut}(A, B) \geq \frac{1}{2} \text{cut}(A^*, B^*)$.

We know $\forall \, u$, $\Delta u \leq 0$ at local optimum.

Then for $u \in A$

$$\sum_{v\in B} w_{uv} \leq \sum_{v\in A} w_{uv}$$

Sum all $u \in A$

$$\underbrace{\sum_{u\in A}\sum_{v\in B} w_{uv}}_{\text{cut}(A,B)} \leq \underbrace{\sum_{u\in A}\sum_{v\in A} w_{uv} = 2W_A}_{*},$$

W_A is the sum of weights in A.

$*$: Since $w_{uv} = w_{vu}$, we are summing twice, hence $2W_A$.

We obtain the same for $u \in B$:

$\text{cut}(A, B) \leq 2W_B$, adding these

$$\cancel{2}\,\text{cut}(A,B) \leq \cancel{2}(W_A + W_B)$$

but $W_A + W_B + \text{cut}(A,B) = \sum_{(u,v)\in E} w_{uv}$, then

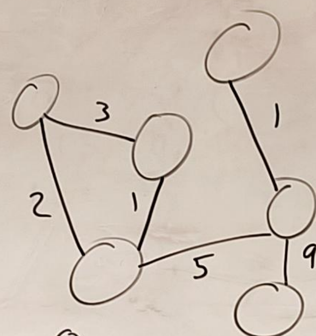
$$\text{cut}(A,B) \leq \sum_{(u,v)\in E} w_{uv} - \text{cut}(A,B)$$

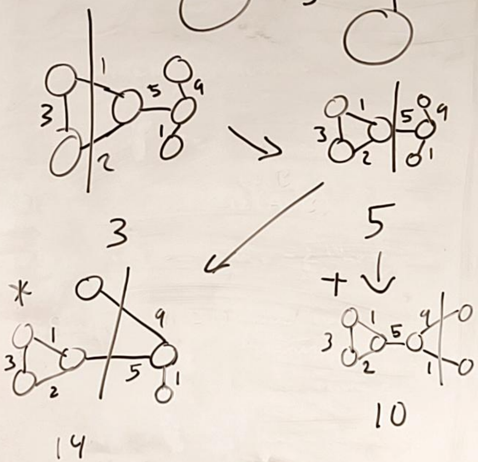
$$2\,\text{cut}(A,B) \leq \sum_{(u,v)\in E} w_{uv}$$

close enough

Task 3

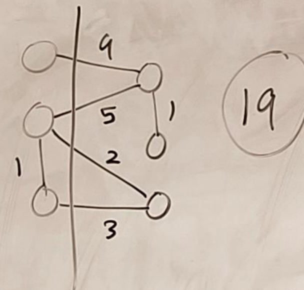
c)





Both of these are local optima.

If the algorithm didn't terminate at * or +, we could continue from * to find global optimum by "switching directions".



d) Trivially, it can be seen that SEARCHMAXIMUMCUT has a time complexity of $O(|V|)$, if we start with a cut like

$A = \emptyset$, $B = V$,

we could have to process $|V|$ nodes.

I have probably missed something...

$\boxed{\text{Task 4}}$

I'm not a mathematician yet...

Let $G=(V,E)$ be a bidirectional, connected graph.

Def its cocycle matroid

$$M^*(G) = (E, \mathcal{I}),$$

$$\mathcal{I} = \{ I \subseteq E \mid \text{the subgraph } (V, E \setminus I) \text{ is connected} \}.$$

Goal: Prove $M^*(G)$ is a matroid.

Plan:

1. recover $M(G)$ primal
2. show $M(G)$ is a matroid
3. profit.

---

1. the mention of a cocycle matroid implies the existance of a cycle matroid $M(G)$. By duality we have $M(G) = (M^*(G))^*$.
Looking online, I find that $I \subseteq E$ is independent in $M(G)$ iff I is contained in a complement of a basis of $M^*(G)$.
The bases of $M^*(G)$ are maximal† I s.t. $(V, E \setminus I)$ is connected $\Rightarrow E \setminus I$ is a spanning tree.
This is reasonable, since removing as many edges as possible from a connected graph s.t. we still have a connected graph should yield a spanning tree.
Since $E \setminus I$ are spanning trees of the bases of $M^*(G)$, we have

$$M(G) = (E, \mathcal{I}),$$

$$\mathcal{I} = \{ I \subseteq E \mid I \text{ is a spanning } \overbrace{\text{forest}}^{\text{independent sets formalities}} \}$$

i.e. there are no cycles. This is the cycle matroid.

2. Use the cryptomorphic circuit axioms of the family $\mathcal{C}$
   - (C1) $\emptyset \notin \mathcal{C}$ no empty cycle Ok
   - (C2) $C_1, C_2 \in \mathcal{C}, C_1 \subseteq C_2 \Rightarrow C_1 = C_2$
     cycles are minimal; start anywhere Ok
   - (C3) $C_1 \neq C_2 \in \mathcal{C}$, $e \in C_1 \cap C_2$, then $\exists\, C_3 \in \mathcal{C}$ s.t. $C_3 \subseteq (C_1 \cup C_2) \setminus \{e\}$ Cycle elimination; Ok
     removing a common edge allows splicing into new cycle.

   These hold in undirected graphs, so $M(G)$ is a matroid.
3. Thus, $M^*(G)$ is also a matroid by duality. ∎

† recall that primal = dual for optimum. I think that is relevant for this maximal magic to convert to primal