Recall the MINVERTEXCOVER problem covered in lectures.

> *Given an undirected graph $G = (V, E)$, return the size of the smallest subset $U \subseteq V$ of vertices such that every edge in E is incident to at least one vertex in U.*

Consider the greedy heuristic: *consider the vertex with the largest degree, add that into the vertex cover, and remove the vertex from the graph and all incident edges.*

We will call this algorithm GREEDYVERTEXCOVER.

(a) Exhibit an instance of G that shows that GREEDYVERTEXCOVER is suboptimal.

Note. *You should provide an instance of a graph, what the greedy heuristic would choose, and what a more optimal solution be; the "more optimal" solution need not be the most optimal solution, it just needs to beat the greedy solution.*

We will show that GREEDYVERTEXCOVER can be made to perform arbitrarily badly; that is, it is not a constant-approximation algorithm. In particular, we will show that it is an $\Omega(\log n)$-approximation.

(b) We define the bipartite graph $G_n = (L \sqcup R, E)$, where L is a set of n vertices. We define R in parts; for each $2 \leq i \leq n$, let R_i denote a set of $\lfloor n/i \rfloor$ vertices, each with degree i. Then $R = R_1 \sqcup R_2 \sqcup \cdots \sqcup R_n$. We define the edges such that all vertices of degree i in L are adjacent to distinct vertices in R.

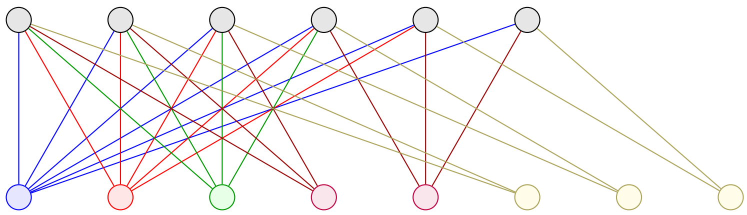


Figure 1: *The graph G_6.*

What does the greedy algorithm choose? What should the optimal vertex cover be?

Hint. *Firstly, figure out what the maximum degree of any vertex in L must be and then use the greedy heuristic to decide what vertices the greedy algorithm picks.*

(c) Let $|\text{GREEDYVERTEX}|$ denote the size of the vertex cover chosen by the greedy algorithm, and let $|\text{OPT}|$ denote the size of the optimal vertex cover. Prove that

$$|\text{GREEDYVERTEX}| \geq n(H_n - 2),$$

where $H_n = \sum_{i=1}^{n} 1/i$ is the nth Harmonic number.

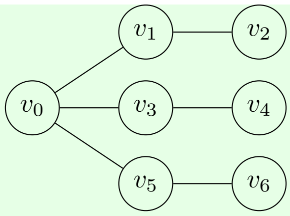
(d) Hence, show that GREEDYVERTEXCOVER is an $\Omega(\log n)$-approximation by proving that

$$\frac{|\text{GREEDYVERTEX}|}{|\text{OPT}|} \geq H_n - 2.$$

Note. $H_n = \log n + \Theta(1)$*; therefore, proving the inequality shows the lower bound approximation.*

Solution.

(a) Consider the following graph.



The greedy algorithm would yield the set $\{v_0, v_2, v_4, v_6\}$ as a solution. This, however, is not the optimal solution, which would be $U = \{v_1, v_3, v_5\}$.

(b) The maximum degree of any vertex in L must be $n-1$. For each $2 \le i \le n$, vertices in R_i will connect to distinct vertices in L. This means that at most $n-1$ connections will be made to the vertices in R because we have $n-1$ distinct degrees from the given range of i values.

Running the greedy heuristic aglorithm will then first pick the vertex in R with degree n as we know there must be one. Each time we remove the incident edges to the vertex, we are reducing the maximum degree of the vertices in L, meaning we will continue to pick the vertices in R. It's clear that $|R| > n$, however, the optimal case should be just choosing the elements in L, with $|L| = n$.

(c) As discussed in the previous part, we must have $|\textsc{GreedyVertex}| = |R|$. So, we have

$$\begin{aligned}
|R| &= \sum_{i=2}^{n} \left\lfloor \frac{n}{i} \right\rfloor \\
&> \sum_{i=2}^{n} \left(\frac{n}{i} - 1 \right) \\
&= n \sum_{i=2}^{n} \left(\frac{1}{i} \right) - (n-1) \\
&= n(H_n - 1) - (n-1) \\
&= n(H_n - 2) + 1.
\end{aligned}$$

Since $|R| > n(H_2 - 2) + 1$, it must be so that $|R| \ge n(H_n - 2)$, which completes the proof.

(d) We know that the optimal solution is $|L| - n$, so

$$\begin{aligned}
\frac{|\textsc{GreedyVertex}|}{|\textsc{opt}|} &\ge \frac{n(H_n - 2)}{n} \\
&= H_n - 2.
\end{aligned}$$

As n grows, the error increases at a rate of $O(H_n) = O(\log n)$, which implies that the algorithm can perform arbitrarily badly.