Solving Vertex Cover for G and k

Algorithm 1 (brute force).

$O(2^n\ n)$

create the power set pow(V)

while no vertex cover found **do**

take a new set S from pow(V)

if S is a vertex cover of size at most k **then**

return yes

return no

Solving Vertex Cover for G and k

Algorithm 2 (search tree/backtracking approach).

inspect the current vertex cover set S

if S is a valid vertex cover and of size at most k **then**

return yes

pick a vertex $x \in V$ \\ we know that x is either in a vertex cover of size k, or it is not

Case 1: include x into S; recursively solve the problem for $G' \leftarrow G\text{-}x$ (that is, vertex x and all its incident edges were removed from G) and $k' \leftarrow k\text{-}1$

Case 2: recursively solve the problem for $G'' \leftarrow G$ and k without allowing x to be part of vertex cover S

if all vertices are processed and no vertex cover of size at most k is found **then**

return no

Solving Vertex Cover for G and k

- Algorithm 3 (bounded search tree/backtracking approach)

- Observation. Given graph $G = (V,E)$ and a vertex cover $V' \subseteq V$ for G

  - For each vertex x in a graph: if $x \notin V'$ then all its a adjacent vertices $\mathrm{N}(x)$ must be in V', that is: $\mathrm{N}(x) \subseteq V'$

- Justification. if neither x nor one of its adjacent vertices is in V', then the edge is not covered. A contradiction.

Solving Vertex Cover for G and k via technique of bounded search trees: Algorithm VC(G,k)

if $E = \varnothing$ and $k \geq 0$ return **yes**

$O(2^k n)$

pick a vertex $x \in V$ with $\deg(x) \geq 1$

$G' \leftarrow G\text{-}x$; $k' \leftarrow k\text{-}1$

$G'' \leftarrow$ from $G\text{-}N(x)$; $k'' \leftarrow k - |N(x)|$

Recursively solve VC(G',k') as long as $k' \geq 0$, and VC(G'', k'') as long as $k'' \geq 0$: **If** VC(G',k') returns no **then return** VC(G'', k'')

return no