BKA
Sheet 8

Due date: 6 July

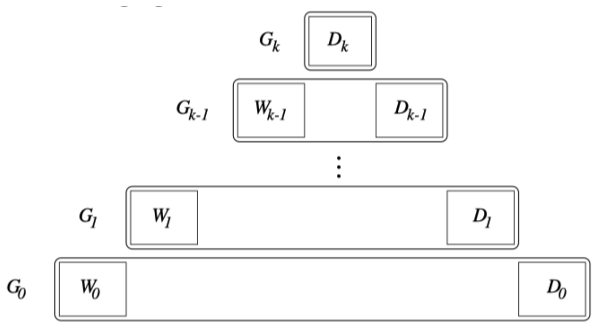
Definition 1 (Weighted set cover)**.** *Given a universe U of n elements, a collection of subsets of U, $S = \{S_1, S_2, \cdots, S_k\}$ and a cost function $c : S \to Q^+$, find a minimum cost subset of S covering all elements of U.*

Definition 2. *Let $w : V \to Q^+$ be the function assigning weights to the vertices of the given graph $G = (V, E)$. A function assigning vertex weights is* degree-weighted *if there is a constant $c > 0$ such that the weight of each vertex $v \in V$ is $w(v) = c \cdot deg(v)$.*

Exercise 1.
Assume OPT is the optimal solution for the weighted vertex cover problem (defined analogously to the weighted set cover problem) over $G = (V, E)$. Show that if the cost function $c : V \to Q^+$ is *degree-weighted*, we have $\sum_{v \in V} w(v) \leq 2 \cdot OPT$.

Definition 3. *Let w be a weight function on the graph G. Remove all vertices of degree zero from G and compute $c = \min_v\{w(v)/deg(v)\}$ (where v goes over the remaining vertices). Then $t(v) = c \cdot deg(v)$ is the* largest degree-weighted function in w*, and $w'(v) = w(v) - t(v)$ is the* residual weight function.

Exercise 2.
Consider the following *layer* algorithm for the vertex set cover problem on $G = (V, E)$. Let $G_0 = G$, and let D_0 be the set of degree-zero vertices in G_0. Compute the largest degree-weighted function in w. Let W_0 be the vertices having zero residual weight, and include these vertices in the vertex cover. Let G_1 be the graph induced on $V - (D_0 \cup W_0)$. Now, repeat the entire process on G_1 with respect to the residual wright function. The algorithm terminates when all vertices are of degree zero; let G_k denote this graph. The process is schematically shown in the following picture. Show this algorithm achieves an approximation guarantee of factor 2 for the vertex cover problem, assuming arbitrary vertex weights.



Let $t_0, \ldots, t_{k-1}$ be the degree-weighted functions defined on graphs $G_0, \ldots, G_{k-1}$. The vertex cover chosen is $C = W_0 \cup \ldots \cup W_{k-1}$. Clearly, $V - C = D_0 \cup \ldots \cup D_k$.

Exercise 3.
What is the approximation factor of the following algorithm for the weighted set cover problem?

Algorithm 2.2 (Greedy set cover algorithm)

1. $C \leftarrow \emptyset$
2. While $C \neq U$ do
   Find the most cost-effective set in the current iteration, say S.
   Let $\alpha = \frac{\text{cost}(S)}{|S-C|}$, i.e., the cost-effectiveness of S.
   Pick S, and for each $e \in S - C$, set $\text{price}(e) = \alpha$.
   $C \leftarrow C \cup S$.
3. Output the picked sets.

Exercise 4.
Give a tight example for the greedy set cover algorithm.

Definition 4 (Shortest superstring)**.** *Given a finite alphabet $\sum$, and a set of n strings, $S = \{s_1, \ldots, s_n\} \subseteq \sum^+$, find a shortest string s that contains each s_i as a substring. Without loss of generality, we may assume that no string s_i is a substring of another $s_j, j \neq i$.*

Exercise 5.
Show the following algorithm gives a two factor approximation for shortest superstring problem.

1. Construct a set cover instance $\mathcal{S}$ as follows:
   - Let the universe of elements be the set of n strings $\{s_1, \ldots, s_n\}$.
   - For $s_i, s_j \in S$ and $k > 0$, if the last k symbols of s_i are the same as the first k symbols of s_j, let σ_{ijk} be the string obtained by overlapping these k positions of s_i and s_j. Let M be the set of all defined σ_{ijk}.
   - For any string, let $set(\pi) = \{s \in S \mid s \text{ is a substring of } \pi\}$.
   - Let the subsets of S in the set cover instance be $\{set(\pi) \mid \pi \in S \cup M\}$.
2. Use the greedy set cover algorithm to find a cover for the instance $\mathcal{S}$.
   Let $set(\pi_1), set(\pi_2), \cdots, set(\pi_k)$ be the sets picked by this cover.
3. Concatenate the string $\pi_1, \pi_2, \cdots, \pi_k$ in any order and output the result