Do not look up research papers on the Web. Work in a group of 2. Submit one homework solution per group.

Total Point: 150

1. Consider the following algorithm for the Steiner Tree Problem

Algorithm

- Let \( \{s_1, s_2, ..., s_{|S|}\} \) be any ordering of the terminals.
- Let \( T \leftarrow \{s_1\} \)
- For \( i = 2 \) to \( |S| \)
  - Let \( P_i \) be the shortest path connecting \( s_i \) to \( T \).
  - Add \( P_i \) to \( T \)

Show that the above algorithm gives a \( O(\lceil \log_2 |S| \rceil) \) approximation factor for the Steiner Tree Problem

Answer.

Let \( c(i) \) denote the cost of the path \( P_i \) used in the \( i \)th iteration to connect the terminal \( s_i \) to the already existing tree. The total cost of our solution is \( \sum_{i=1}^{|S|} c(i) \). Let \( \{i_1, i_2, ..., i_{|S|}\} \) be a permutation of \( \{1, 2, ..., |S|\} \) such that \( c(i_1) \geq c(i_2) \geq ... \geq c(i_{|S|}) \).

Lemma 1. For all \( j \), the cost \( c(j) \leq \frac{2OPT}{j} \), where \( OPT \) is the cost of an optimal solution to the given instance.

Proof. We prove it by contradiction. Suppose, if possible connecting the \( j \)th highest cost terminal \( s_{i_j} \) is more than \( \frac{2OPT}{j} \). This implies, there exist \( j \) terminals each pay more than \( \frac{2OPT}{j} \) to connect to the tree that exists when they are considered. Suppose \( S' = \{s_{i_1}, s_{i_2}, ..., s_{i_j}\} \) denote this set of terminals.

No two terminals in \( S' \cup \{s_1\} \) are within distance \( \frac{2OPT}{j} \) of each other. If some pair \( x, y \) were within this distance, one of these terminals (say \( y \)) must be considered later by the algorithm than the other. Then cost of connecting \( y \) to the tree will be at most \( \frac{2OPT}{j} \) giving a contradiction.

Therefore, the minimum distance between any two terminals in \( S' \cup \{s_1\} \) must be greater than \( \frac{2OPT}{j} \). Since there must be \( j \) edges in any MST of these terminals, an MST must have cost greater than \( 2OPT \). But the MST of a subset of terminals cannot have cost more than \( 2OPT \) (recall the proof that MST on terminal nodes gives a 2-approximation to the Steiner Tree problem). Therefore, we obtain a contradiction. \( \square \)
Given this claim, it is easy to prove the desired claim.

\[
\sum_{i=1}^{|S|} c(i) = \sum_{j=1}^{|S|} c(i_j) \leq \sum_{j=1}^{|S|} \frac{2OPT_j}{j} = 2H_{|S|}OPT
\]

2. Given a complete undirected graph \(G = (V, E)\) with nonnegative edge weights, where edge weights satisfy triangle inequality, and \(k\) colors \(c_1, c_2, ..., c_k\), find an assignment \(\phi\) of colors to vertices such that

- Each vertex is assigned exactly one color.
- Let \(d_r(v)\) be the distance to nearest node from \(v\) that is assigned color \(c_r\). Let \(D_v = \max_{r=1}^k d_r(v)\). The assignment must minimize the maximum \(D_v\) over all \(v\), that is find \(\phi\) such that \(\max_v D_v\) is minimized.

(i) Show the above problem is NP-Hard.
(ii) Obtain a 3-approximation algorithm for the above problem.

\textbf{Answer.}

(i) Easy, reduce K-center.

(ii) Follow the K-center algorithm as described in the class. If the guessed distance \(d\) is correct, then in \(G\), every vertex has at least \(k-1\) neighbors. In \(G^2\), compute the maximal independent set \(I_{G^2} = \{v_1, v_2, ..., v_s\}\). Note that \((v_i, v_j) \notin E\), for \(1 \leq i, j \leq s\). For each \(v_i \in I\), put color \(c_1\) in \(v_i\) and colors \(c_2, c_3, ..., c_k\) in any of its \(k-1\) neighbors.

Every vertex has a path of length at most 2 to some vertex in \(I\). Therefore, every vertex can reach all the \(k\) colors within 3 hops. Therefore, when the guess is correct, the maximum distance traveled by any vertex is at most 3\(d\).

3. Given an undirected graph \(G = (V, E)\), find a spanning tree \(T\) of \(G\) that has maximum number of leaves.

(i) Show the above problem is NP-Hard.
(ii) Consider the following local search heuristic.

- Start with any arbitrary spanning tree \(T\)
- While there are edges \(e \in T\) and \(f \notin T\) such that removing \(e\) from \(T\) and including \(f\) creates a spanning tree with more leaves, \(\text{swap}(e, f)\)
- Return \(T\) when no such improving swaps exist.

Show that the above local search algorithm gives an approximation factor of at most 10.

\textbf{Answer.}

To obtain the above result, first prove the following claims. Let \(n_i\) denote the number of nodes of degree \(i\) in \(T\) and let \(n_{\geq i}\) denote the number of nodes of degree at least \(i\) in \(T\).

(a) Prove for any tree \(T\), \(n_{\geq 3}(T) < n_1(T)\).
Proof. Let $T$ be an arbitrary tree. Then

$$\sum_{v \in V} deg_T(v) = 2(n - 1) \quad (1)$$

Using the notation defined above, we have:

$$\sum_{v \in V} deg_T(v) = \sum_{i=1}^{n-1} in_i \quad (2)$$

$$= n_1 + 2n_2 + \sum_{i \geq 3} in_i \geq n_1 + 2n_2 + 3 \sum_{i \geq 3} n_i \quad (3)$$

Therefore, we have

$$n_1 + 3 \sum_{i \geq 3} n_i \leq 2(n - n_2 - 1) = 2n_1 + 2 \sum_{i \geq 3} n_i - 2$$

or

$$\sum_{i \geq 3} n_i \leq n_1 - 2 \quad \square$$

(b) Define a 2-path to be a maximal (longest) path such that all internal nodes in the path have degree exactly 2 in $T$. Let $n_{2Paths}$ denote the number of such maximal 2-paths. Show $n_{2Paths} < 2n_1(T)$.

Proof. Note that each 2-path is bounded by nodes of degree 1 or at least 3. This implies that $n_{2Paths} \leq n_1 + n_{\geq 3}$. By previous claim, this is at most $2l(T)$, where $l(T)$ denotes the leaves of tree $T$. \square

Use (a) and (b) to establish an approximation factor of at most 10 for the local search algorithm.

Proof. Let $T$ be an arbitrary spanning tree of $G$ and $T'$ a tree output by our algorithm. Recall that we are trying to upper bound the number of leaves of $T$ by the number of leaves of $T'$. Consider now the following partition of the nodes of $T'$ into 3 bins: the first bin contains all the nodes of degree 1, the second bin contains all the nodes of degree 2, and the third bin contains all the nodes of degree $\geq 3$. We count the maximum number of leaves of $T$ that can be in each bin. The first one can have at most $n_1(T') = l(T')$ leaves since this is the size of the bin. Similarly, the third bin can have at most $n_{\geq 3}(T') \leq l(T')$ leaves. Then, to obtain a 10-approximation, it is sufficient to show that the second bin has at most $8l(T')$ leaves of $T$.

To show this we further partition the nodes according to the 2-paths that they belong to in $T'$. Since, the number of 2-paths is at most $2l(T')$, it is sufficient to show that each 2-path can contain at most 4 leaves of $T$.

Suppose, that is not the case. Then, there exists a 2-path $P$ which contain at least 5 leaves of $T$. Denote them by $y_1, y_2, y, y'_1, y'_2$ (see Figure 1). Let $y'$ be a node not in $P$. Let $path_T(y, y')$
denote the unique path from $y$ to $y'$ in $T$. Let $u$ be the node closest to $y$ in $path_T(v, v')$ such that $u$ is not in $P$. Let $w$ be the last node from $P$ on $path_T(y, y')$ before $u$. Note that none of $y_2$ and $y'_2$ belong to $path_T(y, w)$, otherwise they won’t be leaves in $T'$, but $w \in path_T(y, y')$. In other words, $path_T(y, u)$ has to break off from $P$ before $y_2$ or $y'_2$. Now by swapping in $(u, w)$, we can remove at least one edge in $P$ incident on either of $y_1, y_2, y'_1, y'_2$, thus increasing the number of leaves of $T'$—a contradiction.

4. Obtain an FPTAS for the following problem.

Given $n$ positive integers $a_1 < a_2 < .... < a_n$, find two disjoint nonempty subsets $S_1, S_2 \subseteq \{1, 2, ..., n\}$ with $\sum_{i \in S_1} a_i \geq \sum_{i \in S_2} a_i$, such that the ratio

$$\frac{\sum_{i \in S_1} a_i}{\sum_{i \in S_2} a_i}$$

is minimized.

Answer. Easy–extend the proof of FPTAS for Knapsack.