University of Freiburg
Dept. of Computer Science
Prof. Dr. F. Kuhn
S. Faour, M. Fuchs, A. Malyusz

Algorithm Theory
Sample Solution Exercise Sheet 3

Due: Friday, 8th of November 2024, 10:00 am

Exercise 1: Covering Unit Intervals *(5 Points)*

We are given a set X of rational numbers. We want to find a minimum sized set S of unit intervals (i.e., intervals of the form $[a, a+1]$), such that each element in X is covered by at least one of these intervals from S. For example, let $X = \{1, 2.5, 3.1, 5\}$, then the set of unit intervals $S_1 := \{[0,1],[2,3],[3,4],[4.5,5.5]\}$ covers all the elements in X, however, S_1 is not minimal. The minimum sized set contains only 3 intervals, for example $S_2 := \{[0,1],[2.3,3.3],[4.5,5.5]\}$.
Now, consider the following greedy algorithm $\mathcal{A}$:
In the first step $\mathcal{A}$ determines some $a \in \mathbb{Q}$ such that $[a, a+1]$ contains the maximum possible number of elements in X. This interval $[a, a+1]$ is added to S and the covered elements are deleted from X. $\mathcal{A}$ then recurs on the remaining elements and stops when X is empty.

a) Determine why $\mathcal{A}$ does not return an optimal solution. *(2 Points)*

b) Provide an efficient greedy algorithm to solve the problem. Argue why your algorithm is optimal. Use an exchange argument for your reasoning! *(3 Points)*

Sample Solution

a) Let $X = \{1, 1.8, 2.0, 2.2, 2.7, 3.2\}$. Clearly the optimal solution only needs 2 intervals e.g. $[1,2]$ and $[2.2, 3.2]$. The greedy approach $\mathcal{A}$ would however pick an interval that covers $1.8, 2.0, 2.2$ and 2.7 in the first step (as this is the maximum number of elements that can be covered), thus we need two more intervals to cover the remaining 2 values. Hence, it includes more intervals than the optimal solution.

b) The optimal algorithm works as follows:
First, we sort X and take the smallest element, say x_1, to create the first interval $[x_1, x_1+1]$. Now, we delete the elements from X that are covered by this interval and proceed until X is empty.
To show the optimallity, let $x_1, x_2, ..., x_n$ be (the sorted) interval starting points selected by this greedy algorithm and let $y_1, ..., y_m$ be the ones that the optimal solution picks. Let i be the first index where $x_i \neq y_i$. Note that since x_i, by the design of the greedy algorithm, is also part of X we have that $y_i < x_i$, cause otherwise the optimal solution would not cover all elements in X. Because there are no elements smaller than x_i that are uncovered, the optimal solution can just replace $[y_i, y_i+1]$ by $[x_i, x_i+1]$ and still covers the same elements.
One can apply this exchange argument a finite number of times to step-wise change the optimal solution to our greedy solution without increasing the number intervals. Hence, this greedy approach returns an optimal solution.

Exercise 2: Graph coloring *(7 Points)*

We say that a undirected graph $G = (V, E)$ has *degeneracy* k if every subgraph of G (and thus also G itself) has a vertex of degree at most k.

a) Show that given a graph G with *degeneracy* k can be colored with at most $k+1$ colors, i.e., there is a labeling of the nodes of G with 'colors' in $\{1, ..., k+1\}$ such that no neighbors are labeled with the same color. *(3 Points)*

b) There is the class of so-called planar graphs, for which it holds that $|E| \leq 3|V| - 6$. Show that every planar graph can be colored with at most 6 colors.[1] *(4 Points)*
Hint: Try to bound the degeneracy of planar graphs.

Sample Solution

a) First, we calculate an ordering $v_1, v_2, ..., v_n$ of the nodes, such that for all $1 \leq i \leq n$, node v_i has a degree of at most d in the induced subgraph $G[v_i, v_{i+1}, ..., v_n]$. By the definition of degeneracy we know that G has one node with that property. We take this node as v_1 and delete it from the graph. The remaining graph $G[v_2, ..., v_n]$ is a subgraph of G and hence also has a node of degree at most d. This is node v_2. We repeat this until we have constructed the complete order.
We will now color the nodes in reverse direction, i.e., we color in the order $v_n, v_{n-1}, ..., v_1$. Whenever we color some node v_i, we know that v_i has degree at most d in this subgraph, hence even if all d neighbors have a different color, v_i will take the leftover color from $\{1, ..., d+1\}$. This proves the statement.

b) First we show that in every planer graph there exists a node with degree ≤ 5. For that, assume this is false, i.e., all nodes have degree ≥ 6. Then, we can bound the number of edges as follows: $|E| = \frac{1}{2}\sum_{v \in V} deg(v) \geq \frac{1}{2}\sum_{v \in V} 6 = 3|V|$. This is a contradiction to the property of planer graphs that $|E| \leq 3|V| - 6$. Hence, at least one node must have degree ≤ 5.

   Note that since every subgraph of a planar graph is also a planer graph, each subgraph also has at least one node of degree at most 5, and hence planer graphs have degeneracy of at most 5. By the result of a), these graphs are 6-colorable.

Exercise 3: Greedy TSP *(8 Points)*

Consider a symmetrical TSP instance. We have seen in the lecture that the *(greedy) nearest neighbor* approach can be arbitrarily bad. In this task we want to show that this is not true if we have some additional constrains. For that assume that all edges have weight either a or b with $0 < a < b$.

a) Prove that the nearest neighbor algorithm from the lecture is still not optimal. *(1 Point)*

b) Prove that the nearest neighbor algorithm from the lecture produces a TSP tour with a cost of at most a $\frac{a+b}{2a}$ factor from the optimal tour. In other words, show that if NN is the coast of the algorithm and OPT is the cost of the optimal tour that

$$\frac{NN}{OPT} \leq \frac{a+b}{2 \cdot a}.$$

   Hint: Assume that OPT uses exactly k edges with weight a. Try to bound how often edges with weight a are used by NN in dependence of k. *(7 Points)*

Sample Solution

a) We can use the same graph as in the lecture, the 4-clique, let's say with nodes t, u, v, w. Assume every edge has weight a except one edge e.g., (v, w). Clearly, the optimal solution will use 4 edges of weight a. When NN starts at v, it will take an arbitrary a edge, so for example to u, from here it will also take an a edge, say to t and from here we will take the a edge to w. Now, to complete the tour, we are forced to take the b edge. Thus, greedy is not optimal.

[1] It is even known that every planar graph is 4-colorable.

b) The first step is to show the hint. We use the idea of edge marking from the lecture. Let therefore A_{OPT} be the edges of weight a of an optimal TSP tour and let A_{NN} be the edges of our NN algorithm. We call an edge $e' \in A_{NN}$ a marked edge if some incident edge $e \in A_{OPT}$ points to it (as in the lecture). We show that every edge in A_{OPT} points to at least one edge in A_{NN}. Let $e = (u, v) \in A_{OPT}$, let w.l.o.g. the NN tour visits u before v. Then from u there is an a edge that can be picked by NN, thus e points to this edge.

Next step is to show that each $e \in A_{NN}$ is pointed to by at most 2 edges from A_{OPT}. Since a TSP tour is a cycle, there can be at most 2 edges from A_{OPT} at each node.

By these two statements it follows $|A_{NN}| \geq |A_{OPT}|/2 = k/2$. Hence, we have the following costs:

$$OPT = k \cdot a + (n-k) \cdot b = k \cdot (a-b) + n \cdot b$$
$$NN \leq k/2 \cdot a + (n - k/2) \cdot b = \frac{k}{2} \cdot (a-b) + n \cdot b$$

Thus, we derive the statement of the task:

$$\begin{aligned}
\frac{NN}{OPT} &\leq \frac{k \cdot (a-b) + n \cdot b - \frac{k}{2} \cdot (a-b)}{k \cdot (a-b) + n \cdot b} \\
&= 1 - \frac{\frac{k}{2}(a-b)}{k \cdot (a-b) + n \cdot b} \\
&\leq 1 - \frac{\frac{k}{2}(a-b)}{k \cdot (a-b) + k \cdot b} \\
&= 1 + \frac{\frac{k}{2}(b-a)}{k \cdot a} \\
&= 1 + \frac{b-a}{2a} \\
&= \frac{a+b}{2a}
\end{aligned}$$