# Assignment 4

## Algorithms: Correctness and Complexity

## Due Nov 30th at 10:30am

**Q0 [15]** Design a greedy algorithm for the following problem.

We are given a finite set of points $P$ on the real number line $P \subset \mathbb{R}$. And a finite set of intervals $S \subset \mathbb{R} \times \mathbb{R}$. We say each interval $(a, b)$ **covers** a point $p$ if $a \leq p \leq b$. We wish to find a smallest subset of $S$ that covers every point in $P$.

(a) [5] Show that this is an example of the schematic optimal subset selection problem which is in the notes.

(b) [5] Create a greedy algorithm for it by reifying the schematic greedy algorithm for optimal subset selection which is in the notes. Be clear about how you choose to define "suitable" and "best suitable". Note that greedily picking the interval that covers the most uncovered points does not work. Try working left to right.

(c) [5] Use a cut and paste (fairy godmother) argument to show your algorithm is correct.

---

**Q1 [20]** Suppose we have a context free grammar $G = (A, N, P, n_{\text{start}})$ in which every production is of the form

- $n \to a$ with $a \in A$ or
- $n \to p\ q$ with $p, q \in N \cup A$.

Notice that, from any nonterminal, we can only derive nonempty strings.

Given a string nonempty $x \in A^*$, we want to know the height of the shortest parse tree for $x$ with $n_{\text{start}}$ at its root.[1] Use $\infty$ to represent that $x$ can not be derived from $n_{\text{start}}$. Hint: Start by generalizing the problem to one of calculating the height of the shortest parse tree for $x[i, ..k]$ with its root labelled by $r$, where $i$ and $k$ are any natural numbers with $0 \leq i < k \leq x.\text{length}$ and $r$ is any member of $N \cup A$.

For example suppose our grammar has an alphabet of $A = \{1, 0\}$ and a nonterminal set of $\{\text{H}, \text{G}\}$. The start nonterminal $n_0$ is H. The productions are

$$\begin{aligned} \text{H} &\to 1\ \text{G} \\ \text{H} &\to \text{H}\ \text{G} \\ \text{H} &\to 1 \\ \text{G} &\to 0\ \text{H} \end{aligned}$$

[1] The **height** of a tree is the maximum number of edges along any traversal from the root to a leaf. A **parse tree** for a grammar $(A, N, P, n_{\text{start}})$ is a tree in which the children of each node are ordered and whose nodes are labelled by elements of $A \cup N$ . If a node is labelled by a member of $A$, it must have no children. It is required that, for each $n \in N$ and each node $p$ labelled by $n$, if the sequence of labels of $p$'s children is $\alpha$ then $n \to \alpha \in P$. A **parse tree for** $x$ is a parse tree whose nodes are labelled by members of $A$ from left to right form the sequence $x$.

The string 1010101 has at least two parse trees with root labeled by H:

$$\mathrm{H}\left(1, \mathrm{G}\left(0, H\left(1, \mathrm{G}\left(0, \mathrm{H}\left(1, \mathrm{G}\left(0, \mathrm{H}\left(1\right)\right)\right)\right)\right)\right)\right)$$

and

$$\mathrm{H}\left(\mathrm{H}\left(1, \mathrm{G}\left(0, \mathrm{H}\left(1\right)\right)\right), \mathrm{G}\left(0, \mathrm{H}\left(1, \mathrm{G}\left(0, \mathrm{H}\left(1\right)\right)\right)\right)\right)$$

These have heights 7 and 5 respectively.

(a)[5] Design an inefficient recursive procedure for the (generalized) problem. Be sure to give the contract for the procedure.

(b)[5] Design a top-down dynamic programming algorithm.

(c)[5] Design a bottom-up dynamic programming algorithm.

(d)[5] Design an efficient algorithm to compute a parse tree of minimum height for a string $x$, if there is one.

---