Purdue University — S. Hambrusch - C. Guerra
CS 381 — Fall 1999

Solution Sketches for Assignment 9

1) (i) Design an efficient and simple algorithm to check whether a given text of length n contains a string of 50 or more consecutive blanks. Analyze your algorithm. If you are not using KMP, state why not. If you are using KMP, state why you don't think there is a simplier solution.

Scan the text. Let i be the position of the character being examined; starting at i compare each character with the blank character and increment a counter j, until either a mismatch is found or the end of the text is reached. In the latter case no string of 50 or more consecutive blanks was found. In the former case, if $j \geq 50$ a string of 50 or more blanks was found, otherwise start over again after incrementing i by j+1.

The time complexity of this algorithm is $O(n)$, since every character is examined exactly once. The KMP algorithm has the same $O(n)$ time complexity as the above algorithm, but is complicated.

(ii) Give a pattern of length 8 beginning with A and using only characters of the alphabet $\Sigma = \{A, B, C\}$ that would have the following fail indexes for the KMP algorithm: 0 1 1 2 3 4 2 2.

ABABAAAA

2) Let T and P be two sequences $t_1 t_2 \cdots t_n$ and $p_1 p_2 \cdots p_k$ of characters, $k \leq n$. Sequence P is a subsequence of T if there exists a sequence of indices $i_1 < i_2 < \cdots < i_k$ such that for all j, $1 \leq j \leq k$, we have $t_{i_j} = p_j$. For example, for T = A B R A C A D A B R E and P = A R C A D E, P is a subsequence of T.
Design an $O(n)$ time algorithm to determine whether P is a subsequence of T. Argue the correctness of your algorithm.

Starting at the beginning of the two strings, compare characters in P and T one at the time, until the end of one string is reached. Let t_i and p_j the characters being compared. If $t_i = p_j$ then advance both indeces, otherwise ($t_i \neq p_j$) advance i only. If the end of P is reached, then P is a subsequence of T. If the end of T is reached, P is not a subsequence of T.

Correctness: Suppose that the algorithm has found that $p_1 p_2 \cdots p_{j-1}$ is a subsequence of $t_1 t_2 \cdots t_{i-1}$, $0 \leq j < k, 0 \leq i < n$. (The empty string is a subsequence of

the empty string). Then the algorithm looks for the first occurrence of the p_j in the string $t_i \cdots t_n$, $j < k, i < n$. If P is a subsequence of T, then it is certainly found in this way.
Matches of characters of P and T could also be obtained in some cases by different choices of the next occurrence of p_j in T, as for instance when there are consecutive occurrences of p_j starting at t_i, but choosing the first occurrence always gives a solution if there is one.

3) (i) Define the decision version of the Hamiltonian Cycle (HC)
Given an undirected n-vertex graph G, decide whether there exists a simple cycle of length n. (See also page 552 of textbook.)

(ii) Define the optimization version of the HC problem.
Given an undirected n-vertex graph G, determine the longest simple cycle in G. Hence, the optimization lies in the length of the cycle. If the length is n, G contains a Hamiltonian cycle.

(iii) The Hamiltonian Cycle problem is known to be NP-complete. Give a brief explanation what this means.
It means that HC is in the class NP, that there exists no known polynomial time algorithm solving the HC problem, and that the existence of a polynomial-time algorithm implies a polynomial-time algorithms for every problem in NP (this would make P=NP).

(iv) Show that the existence of a polynomial-time algorithm for the HC decision problem implies the existence of a polynomial time algorithm for the HC optimization problem.
Let $G = (V, E)$ be an undirected graph and let HC-Dec be the polynomial-time algorithm solving the decision version of the HC problem. We first invoke HC-Dec on G: if the answer is "no", we report that G contains no Hamiltonian cycle; if the answer is "yes", we find a Hamiltonian cycle as follows.

We invoke HC-Dec at most m times, once in each of at most m iterations. Let $e_1, \ldots, e_m$ be the m edges of G. At the i-th iteration, we remove edge e_i from the current graph. The "current graph" is initially graph G. During the iterations, the current graph changes as edges are removed or are marked as belonging to the Hamiltonian cycle. We invoke HC-Dec on the current graph. If the answer is that the current graph contains no Hamiltonian cycle, we put edge e_i back into the graph and mark it as an edge of the cycle. If the answer is that the current graph contains a Hamiltonian cycle, we do not put edge e_i back into the graph. Observe that the current graph contains a cycle not using edge e_i. The algorithm terminates when the current graph contains n edges - these are the edges forming a Hamiltonian cycle. Since we are invoking HC-Dec at most m times, the resulting algorithms has polynomial running time as long as HC-Dec has polynomial running time.

(v) The degree-restricted spanning-tree problem is (DRST) defines as follows: Given is an undirected graph G and an integer k. Determine whether G contains a spanning tree T such that every vertex of T has degree at most k. Show that DRST is NP-complete.

The first step is to show that DRST is in NP. The certificate/solution is a spanning tree T. To verify that T is indeed a solution we need to check that it is a spanning tree of the graph and that every vertex in T has degree at most k. Using graph algorithms seen earlier, this can be done in polynomial time.

Solution 1: a reduction from HC to DRST.
Choose HC as the NP-complete problem and show $HC \leq_P DRST$. Let $G = (V, E)$ be the input for the HC problem with $V = \{1, \ldots, n\}$. Let $G' = (V', E')$ and k be the input for the DRST problem. Set $k = 2$ and

$$V' = V \cup \{1', s, t\}$$

$$E' = E \cup \{(1', u) | (1, u) \in E\} \cup \{(1, s), (1', t)\}.$$

Graph G' can be constructed from G in polynomial time. To show that this is a correct reduction, we need to show that G has a HC if and only if G' has a spanning tree of degree at most 2.

To show this, assume first that G has a HC. Let $v_1 = 1, v_2, \ldots, v_n, v_1 = 1$ be this cycle. Then, the edges $(s, 1), (1, v_2), (v_2, v_3), \ldots, (v_n, 1'), (1', t)$, form a tree in which every vertex has degree at most 2 (it is a path from s to t).

Assume now that G' contains a spanning tree with degree at most 2. By the construction of G', s and t are the only two leaves of the tree. The tree consists of a path from s to t containing $n + 3$ vertices. Let $s, 1, w_2, w_3, \ldots, w_n, 1', t$ be this path. By construction of G' from G, it follows that G contains the edge $(w_n, 1)$ (note that 1 and 1' are incident to the same set of vertices). Hence, G contains the cycle $1, w_2, w_3, \ldots, w_n, 1$ which is a Hamiltonian cycle.

Solution 2: a reduction from HP to DRST.
Choose HP as the NP-complete problem and show $HP \leq_P DRST$. Let $G = (V, E)$

be the input for the HP problem. We now simply invoke the algorithm for the DRST problem with graph G and $k = 2$ as input. If G contains a Hamiltonian path, G contains a spanning tree in which every vertex has degree at most 2. On the other hand, if G contains such a spanning tree, it also contains a Hamiltonian path.