Purdue University
CS 381 | Hambrusch
Spring 2015

Assignment 6

Due: Thursday, April 9, 2015 (hand in before class)

1) (15 pts) Show that each problem listed has an $\Omega(n \log n)$ lower bound.

(i) Given are n elements $a_1, a_2, \ldots, a_n$ and a value $\epsilon \geq 0$. Determine whether there exist two elements a_i and a_j such that $a_i - a_j = \epsilon$.

(ii) Given are n blocks of length $b_1, b_2, \ldots, b_n$ and n rods of length $r_1, r_2, \ldots, r_n$. You want to assign to every rod r_i to a unique block of length $b_{f(i)}$ such that $\sum_{i=1}^{n} |b_{f(i)} - r_i|$ is a minimum over all possible assignments. The algorithm returns the pairs $(i, f(i))$, $1 \leq i \leq n$.

2) (20 pts.) Let $G = (V, E)$ be a connected, undirected n-vertex, m-edge weighted graph. Assume edge weights are distinct.

(i) Prove or disprove the following statement: Let C be any cycle in G and let e be the edge of minimum weight on cycle C. Then, edge e is an edge in the minimum-cost spanning tree of G.

(ii) Assume k edges $e_1, e_2, \ldots, e_k$ are identified as *major edges* and every minimum cost spanning tree must include the k major edges, $k \leq \log n$. You can assume the k edges form a forest. Describe a minimum-cost spanning tree algorithm that includes the k major edges and selects additional edges of minimum cost. Give the overall time bound when the graph is represented by adjacency lists and the k major edges are given in a linked list.

(iii) For a given path P between two vertices u and v let max_weight(P) be the largest edge weight on path P. For every pair of vertices u and v, let min_cap(u, v) be the minimum of all max_weight(P_i) entries taken over all paths P_i from u to v in graph G; i.e., min_cap$(u, v) = \min_{P_i}\{$max_weight$(P_i)\}$.

Let T be the minimum cost spanning tree of G. Prove that for every pair of vertices u and v, the maximum weight on the path from u to v in T is equal to min_cap(u, v).

3) (15 pts.)

(i) Give an example of a string T of length n and a pattern P of length m which forces the brute-force pattern matching algorithm to have a running time of $\Omega(nm)$. Explain why.

(ii) How many prefixes of $P = AAABBAAA$ are also a suffix of P? Show all of them.

(iii) Determine the failure function for the pattern $CGTACGTTCGTAC$. Explain how the entries are generated.