CS325 Winter 2013: HW2

Due Jan 25th in class

Hand in Instructions Read the guidelines for written assignments on the course web site. You are highly encouraged to work in groups of up to three.

1. Given two sorted arrays $a[1, ..., n]$ and $b[1, ..., n]$, given an $O(\log n)$ algorithm to find the median of their combined $2n$ elements. (Hint: use divide and conquer).

2. Prove the following statement by induction: In any full binary tree, the number of leaves is exactly one more than the number of internal nodes.

   Definition: A node is a leaf if it has no children; otherwise, it is an internal node. A full binary tree is a binary tree whose node is either a leaf node or an internal node with exactly two children.

3. **Interval scheduling.** We are given a set of requests for using a resource. Each request i specifies a starting time $s(i)$ and an end time $f(i)$. The resource can only accommodate one request at a time. If two requests overlap in time, they are incompatible and cannot be both fulfilled. The goal is to identify a maximum subset of compatible requests. One possible greedy strategy is to select at each step the request that is compatible with the maximum number of the remaining requests. *Will this greedy strategy lead to an optimal solution? If so, provide a proof. If not, provide a counter example.*

4. You and your friends are taking a long hiking trip of L miles, along which there are n camping sites located at distances $x_1, x_2, ..., x_n$ respectively from the start of the trip. You can hike at most d miles per day, by the end of which you must stop and camp for the night. You need make a valid trip plan that takes the minimum number of camping stops. The plan should specify which camping sites to use, and it is only **valid** if any two consecutive stops are no more than d miles apart. Your friend proposed the following strategy: each time you come to a camp site, check whether you can make it to the next site before the end of the day (i.e., before finishing the d miles quota for the day. We assume this can always be determined correctly), if so, keep hiking. If not, stop for the night.

   This is in fact a greedy algorithm, which simply choose to hike as long as possible each day. Prove that this greedy algorithm achieves the optimal solution, i.e, it uses the minimum number of stops. (Hint: construct a proof that is similar to the interval scheduling proof, which shows that the greedy algorithm stays ahead.)