CS 473: Fundamental Algorithms, Fall 2014

Discussion 8

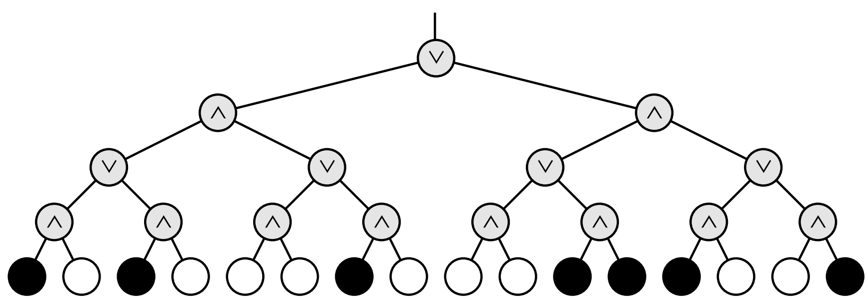
October 21 / 22, 2014

8.1 Game tree evaluation.

Death knocks on your door one cold blustery morning and challenges you to a game. Death presents you with a complete binary tree with 4^n leaves, each colored either black or white. There is a token at the root of the tree. To play the game, you and Death will take turns moving the token from its current node to one of its children. The game will end after $2n$ moves, when the token lands on a leaf. If the final leaf is black, you die; if it's white, you will live forever. You move first, so Death gets the last turn.

(a) Describe and analyze a deterministic algorithm to determine whether or not you can win.

(b) Unfortunately, Death won't let you look at every node in the tree. Assume that you have at least 1/2 chance of being able to win and describe a *randomized* algorithm that determines whether you can win in $\Theta(3^n)$ expected time.



8.2 Find kth smallest number.

This question asks you to design and analyze a *randomized incremental* algorithm to select the kth smallest element from a given set of n elements.

In an *incremental* algorithm, the input consists of a sequence of elements $x_1, x_2, \ldots, x_n$. After any prefix $x_1, \ldots, x_{i-1}$ has been considered, the algorithm has computed the kth smallest element in $x_1, \ldots, x_{i-1}$.

Any incremental algorithm can be *randomized* by first randomly permuting the input sequence, with each permutation equally likely.

(a) Describe an incremental algorithm for computing the kth smallest element.

(b) How many comparisons does your algorithm perform in the worst case?

(c) What is the expected number (over all permutations) of comparisons performed by the randomized version of your algorithm? (Hint: When considering x_i, what is the probability that x_i is smaller than the kth smallest so far?) You should aim for a bound of $n + O(k^2 \log(n/k))$. Revise (a) if necessary in order to achieve this.

8.3 Let G be a graph. A cut of G is a partition (A, B) of the vertices. The size of a cut (A, B) is the number of edges $(u, v) \in E(G)$ such that $u \in A$ and $v \in B$.

Prove that the expected size of a random cut is $E(G)/2$.

8.4 A data stream is an extremely long sequence of items that you can only read in one pass (i.e., each item can only be read once).

(a) Describe and analyze an algorithm that chooses one element uniformly at random from a data stream without knowing the length of the stream in advance. Your algorithm should spend $O(1)$ time per item and use $O(1)$ space.

(b) Describe and analyze an algorithm that chooses k elements uniformly at random from a data stream without knowing the length of the stream in advance. Your algorithm should spend $O(1)$ time per item and use $O(k)$ space.