CMPS 2200 Introduction to Algorithms – Fall 15

11/18/15

10. Homework

Due **11/30/15** at the beginning of class

1. **Union-Find (6 points)**

```
for(i=1; i<=10; i++) x[i]=MAKE-SET(i);
UNION(x[1],x[2]); UNION(x[4],x[5]); UNION(x[7],x[8]); UNION(x[9],x[10]);
UNION(x[5],x[9]); UNION(x[3],x[6]); UNION(x[6],x[8]);
UNION(x[5],x[8]);
UNION(x[2],x[10]);
FIND-SET(x[6]);
```

   Assume an implementation of the Union-Find data structure with a disjoint-set forest with union-by-weight and path compression.

   Show the data structure after every line of code. What is the answer to the `FIND-SET` operation?

2. **Ackermann (2 points)**
   What is the value of $\alpha(10^{500})$? Justify your answer.

3. **Maximum in an array (7 points)**
   An array $A[0..n-1]$ contains n distinct numbers that are randomly ordered, with each permutation of the n numbers being equally likely. The task is to compute the expected value of the index of the maximum element in the array.

   (a) (1 point) Describe the sample space.

   (b) (1 point) Describe the random variable of interest

   (c) (1 point) Consider an example array of $n = 6$ numbers. Consider two different orderings of the numbers in this array, and for each of these orderings provide the value of the random variable.

   (d) (2 points) Now consider an arbitrary n again. What is the probability that the maximum of the array is contained in the first slot? And what is the probability that it is contained in the second slot?

   (e) (2 points) Use the following definition of an expected value to compute the expected value of your random variable.

$$E(X) = \sum_{x=0}^{n-1} P(X = x) * x$$

   Note that $P(X = x)$ is short for $P(\{s \in S \mid X(s) = x\})$, or in other words this is the probability that the random variable X takes one specific value x.

Flip over to next page ⟹

4. **Decision tree (6 points)**
   Draw the decision tree for Mergesort for an array $A[0..2]$ of $n = 3$ elements. Note that the first split is $A[0] : A[1..2]$. Annotate the decision tree with comments indicating the part of the algorithm that a comparison belongs to.

5. **Lower bound for comparison-based searching (5 points)**
   Consider the problem of searching for a given key in a sorted array of n numbers. Use a decision tree to show a lower bound of $\Omega(\log n)$ for any comparison-based search algorithm. *(Hint: What should be stored in the leaves as the output of the search algorithm?)*