CPS 130 Final
Fall 1999
9:00am-12:00p.m, Tuesday Dec 14th
Closed book exam

Do any 7 questions of your choice

Problem 1 (20 points)
Indicate whether the following statements are true or false. Also mention in one sentence the reason behind your answer.

1. Both Mergesort and Heapsort are stable sorts. (A sorting method is stable if equal elements remain in the same relative order in the sorted sequence as they were in originally.)
2. Quicksort performs better than Insertion Sort in all cases.
3. A splay tree is a balanced binary search tree.
4. There always exists an augmenting path for a non-maximum maximal matching, and that augmenting path is unique.
5. The strongly connected components of a directed graph can be computed in linear time.
6. Given a graph $G = (E, V)$, let $e$ be the third smallest edge in $G$ according to the weight function $w$. We can always construct a Minimum Spanning Tree containing $e$, if we follow the ”greedy” strategy.

Problem 2 (20 points)
Consider a rectangular array. Sort the elements in each row into increasing order. Next sort the elements in each column into increasing order. Prove that the elements in each row remain sorted. (Hint: prove by contradiction.)

Problem 3 (20 points)
Let $S$ be a set of $n$ integers. Assume you can perform only addition of elements of $S$ and comparisons between sums. Under these conditions how many comparisons are required to find the maximum element of $S$? (Hint: we know that if $a + c < b + c$, then $a < b$.)
Problem 4 (20 points)

Let $A$ and $B$ be two sorted arrays of $n$ elements each. We can easily find the $k$th smallest element in $A$ in $O(1)$ time by just outputting $A[k]$. Similarly, we can easily find the $k$th smallest element in $B$. Give an $O(\log k)$-time algorithm to find the $k$th smallest element overall — i.e., the $k$th smallest in the union of $A$ and $B$. You may assume there are no duplicate elements.

Problem 5 (20 points)

Let $S$ be an array of $n$ integers for which we are told that one integer $x$ occurs more than $\lceil n/3 \rceil$ times in $S$. Describe an $O(n)$-time algorithm to find $x$. (Hint: use the linear time selection algorithm.)

Problem 6 (20 points)

Consider a binary search tree in which each node $v$ contains a key as well as an additional value called 'addend'. The addend of node $v$ is implicitly added to all keys in the subtree rooted at $v$ (including $v$). Let $(\text{key}, \text{addend})$ denote the contents of any node $v$.

For example, the following tree contains the elements $\{5, 6, 7\}$:

```
(4,2)
  /
(3,0)  (4,1)
```

a) Which elements does the following tree contain?

```
(6,4)
  /
(0,2) (15, -3)
  /
(-1,0) (2,0) (4,6) (10,7)
  /
(0,1) (7,0)
```

b) Let $h$ be the height of a tree as defined above. Describe how to perform the following operations in $O(h)$ time:

- FIND($x$, $T$): return YES if element $x$ is stored in tree $T$
- INSERT($x$, $T$): inserts element $x$ in tree $T$
- PUSH($x$, $k$, $T$): add $k$ to all elements $\geq x$
Problem 7 (20 points)

a) Draw a depth-first search forest. (Suppose we start from node $a$.)

b) Find the strongly connected components.

Problem 8 (20 points)

A cone-graph is a directed graph with non-negative weights with the vertices arranged on a square grid as follows (weights not shown):

The vertex with in-degree 0 is called the source and the vertex with out-degree 0 is called the sink. The degree of a cone-graph is defined as the number of vertices along one of the sides and thus a cone-graph of degree $k$ has $k^2$ vertices in total. The degree of the cone-graph shown above is 5.

In the following problems all time bounds should be given in terms of $k$ and it can be assumed that the graph is represented such that each vertex can access its neighbors in constant time.

a) How long would it take Dijkstra’s algorithm to find the length of the shortest path from the source to the sink?

b) Describe a more efficient algorithm for finding the length of the shortest path from the source to the sink. (Hint: take advantage of the specialty of the cone-graph.)
Problem 9 (20 points)

A bike-chain graph is an undirected weighted graph consisting of simple cycles called links. Two links have a single vertex in common. Such a vertex is called a pin. Pins have degree four while all other vertices have degree two. The following is an example of a bike-chain with three links and two pins.

a) Assume that a bike-chain graph $G$ has $k$ links, $n$ vertices, and $m$ edges. Give $m$ as a function of $k$ and $n$.

b) Describe and analyze an efficient algorithm for computing the minimal spanning tree of $G$.

Problem 10 (20 points)

Let $A[1..n]$ be an array of $n$ numbers, some of which may be negative. Give an $O(n)$-time algorithm to find the range $[i, j]$ such that the sum $A[i] + \ldots + A[j]$ is maximized. (For convenience, if $j < i$, we define the sum to be zero.) For example, if $A = [2, -1, 3, -5, 2]$, then the optimal range is $[1, 3]$. (Hint: try to come up with an exponential-time recursive algorithm based on the idea that either you use $A[n]$ or you don’t, and then look at how to speed it up.)

Extra Credit (20 points)

A $k$-coloring of an undirected graph $G = (V, E)$ is a function $c : V \leftarrow \{1, 2, \ldots, k\}$ such that $c(u) \neq c(v)$ for every edge $(u, v) \in E$. In other words, the numbers $1, 2, \ldots, k$ represent the $k$ colors, and adjacent vertices must have different colors. The graph-coloring problem is to determine the minimum number of colors needed to color a given graph.

a) Argue that a tree can be 2-colored. (Hint: prove by induction)

b) Give an efficient algorithm to determine a 2-coloring of a graph if one exists.