

Graduate Comprehensive Exam: Data Structures and Algorithms (Spring 2005)

Answer all questions on the exam. You may use the back for additional space. Total: 100 points. Good Luck.

1. **(15 points)** Using big-O notation, estimate the running time of `proc(N)` (explain your answer and show your steps):

```
void proc(int x)
{
    if (x > 1)
    {
        for (int i = 0; i < x; i++)
            { /* a constant-time operation here */ }
        proc(x/2);
    }
}
```

2. **(15 points)** Selection sort

- (a) Analyze the best and worst cases for total number of comparisons
- (b) Analyze the best and worst cases for total number of moves/swaps

3. **(15 points)** Consider this sequence of integers: 5, 3, 2, 1, 6, 4, and perform Heapsort. Show the content of the heap after an element is sorted (and removed) from the heap.

4. **(5 points)** Consider Hawaii wants to connect its various islands with bridges so that residents/tourists can travel by car from any island to any other island. Given the approximate cost of building each bridge (between any two islands), help Hawaii find out the cheapest way to build the bridges. How do you solve this problem with a graph?

For the following questions, you may use pseudocode (with sufficient details) or a high-level programming language (like C, C++, or Java) to *write* a function.

5. **(25 points)** Write a function that takes a binary tree as input and prints the reverse-level order of the nodes of the tree. For example, if the tree's height is 4 and the root is at level 1, your function would print nodes at level 4, followed by level 3, and so on. Make sure you provide the interface of the tree you are using in your traversal function.

6. (25 points)

- (a) Write a function `NumWithTwoChildren`. This function is passed a pointer to the root node of a binary tree and returns the number of nodes in the tree that contain two children. You may not add any global or static variables, or alter the function signature.
- (b) What is the running time of your algorithm?