University of Nevada, Las Vegas Computer Science 477/677 Spring 2021

Practice for Final Examination: Part I

This portion of the practice final is 205 points.

1. True or False. [5 points each]

   (a) _____ Computers are so fast today that complexity theory is only of theoretical, but not practical, interest.

   (b) _____ There is a Las Vegas version of quicksort whose expected time complexity is $O(n \log n)$.

   (c) _____ There is no comparison-based sorting algorithm for 5 items which uses fewer than 7 comparisons in the worst case.

   (d) _____ There is a deterministic algorithm which finds the median of a set of n numbers in $O(n)$ time.

2. Fill in the blanks. [5 points each blank.]

   (a) What is the worst-case time complexity of binary search on a sorted array of size n? (Use Θ notation.)

   (b) Heapsort is a sophisticated version of which one the following three simple algorithms – selection sort, insertion sort, bubblesort.

   (c) Any comparison-based sorting algorithm for 5 items must use _____ comparisons in the worst case. (Exact answer, please.)

   (d) Treesort is a sophisticated version of which one the following three simple algorithms – selection sort, insertion sort, bubblesort.

   (e) Name two well-known $\Theta(n \log n)$ time sorting algorithms.

   ---

3. Solve the recurrences. Give asymptotic answers in terms of n, using either Θ, Ω, or O, whichever is most appropriate. Use the master theorem or the anti-derivative method in each case. You may need to use substitution. [10 points each.]

   (a) $F(n) >= 4F(n/2) + n^2$.

(b) $G(n) = G(n-1) + n^2$

(c) $H(n) = 2H(\sqrt{n}) + \log n$.

(d) $J(n) < J(n - \sqrt{n}) + 1$. (Hint: you do not need to use substitution.)

(e) $K(n) <= 2K(n/4) + \sqrt{n}$ (Hint: you do not need to use substitution.)

(f) $L(n) = L(\log n) + 1$

(g) $G(n) >= G(\sqrt{n}) + 1$

(h) $F(n) = F(n/2) + F(n-1) + 1$ (Hint: Of course there is a solution, but I know of no way to express the solution in closed form. If anyone can come up with any insight on this problem, I'll be grateful.)

4. [15 points] Consider the following C++ code.

```
void martha(int n)
 {
  for(int i = 1; i < n; i = 2*n)
  cout << "Hello, George!" << endl;
 }
void george(int n)
 {
  for(int i = n; i > 1; i = i/2)
   martha(i);
 }
```

What is the time complexity of `george(n)`?

5. [30 points] Walk through polyphase mergesort, where the input file is as given below.

ABQXDFMGLKYT

6. [20 points] The following code implements an algorithm we've discussed in class, on an array A. What algorithm does the code implement?

```
void swap(int&x,int&y)
 {
  int temp = x;
  x = y;
  y = temp;
 }

void main()
 {
  for(int i = 0; i < n; i++)
   for(int j = i+1; j < n; j++)
    if(A[i] > A[j]) swap(A[i],A[j]);
 }
```

7. [20 points]

```
int product(int a, int b)
 {
  assert(b >= 0);
  int c = a;
  int d = b;
  int total = 0;
  while(b > 0)
   {
    if(b%2) total = total + a;
    a = 2*a;
    b = b/2;
   }
  return total;
 }
```

(a) What does this function do?

(b) What is the loop invariant of the while loop?