MCS 401 – Computer Algorithms I
Spring 2016
Problem Set 1

Lev Reyzin

Due: 2/1/16 by the beginning of class

Instructions: Atop your problem set, write your name and whether you are an undergraduate or graduate student. Also write the names of all the students with whom you have collaborated on this problem set.
Important note: Problems labeled "(**U**)" and "(**G**)" are assigned to undergraduate and graduate students, respectively. Undergraduate students can get a small bonus for solving the graduate problems. Graduate students are encouraged to solve the undergraduate problems for practice.

1. [10 pts] Consider an execution of the Gale-Shapley algorithm on n men and n women.

(**U**) If the algorithm makes exactly $n+1$ proposals before terminating, how many men are assigned their first choice women? Prove your answer correct.

(**G**) Is it possible for two different men to both be assigned their respective last-choice woman? Prove your answer correct.

2. [10 pts] Consider extending the stable marriage problem to the case of fellowship programs in a given specialty (which doctors can complete as additional training after their residency). Each of m doctors ranks all the fellowship programs. Each of n programs ranks its applicants. Each doctor can be admitted to at most one fellowship program, but each program has a different number of slots to fill in its class. Assume that there may be more applicants than total slots, so some doctors might not be assigned to any fellowship.
An assignment of doctors to programs is stable if neither of the following two instabilities exist:

- There are doctors d and d', and a program p, so that d is assigned to p, and d' is assigned to no program, and p prefers d' to d.
- There are doctors d and d', and programs p and p', so that d is assigned to p, and d' is assigned to p', and p prefers d' to d, and d' prefers p to p'.

Prove that a stable assignment of doctors to fellowship programs always exists.

3. [10 pts] Arrange the following functions in ascending order of asymptotic growth rate; that is, if function $g(n)$ immediately follows function $f(n)$ in your list, then it should be the case that $f(n)$ is $O(g(n))$: $2^{\sqrt{\log n}}, 2^n, n^{4/3}, n(\log n)^3, n^{\log n}, 2^{2^n}, 2^{n^2}$. Justify your answer.

4. **[10 pts]** Consider the following problem. You are given array X consisting of integers $X[1], X[2], \ldots X[n]$, and you need to output an $n \times n$ matrix Y in which the entries $Y[i,j] = X[i] + X[i+1] + \ldots + X[j]$ for $i < j$ (for $i \geq j$, $Y[i,j]$ may contain any value). Consider the following algorithm for this problem.

Algorithm 1 Compute Y (given input X, an array of n integers)

```
initialize Y to an n × n matrix of 0s
for i = 1 to n do
    for j = i + 1 to n do
        for k = i to j do
            Y[i, j] = Y[i, j] + X[k]
        end for
    end for
end for
return Y
```

i. Give a function $f(n)$ that bounds the worst-case running time of Algorithm 1 as $\Theta(f(n))$. Note: to do this, you must show that the worst-case running time of Algorithm 1 is both $O(f(n))$ and also $\Omega(f(n))$.

ii. Using the function $f(n)$ from part i., design an algorithm with running time $o(f(n)$. (This is called little-O notation. In other words, give an algorithm with running time $g(n)$, where $\lim_{n\to\infty} g(n)/f(n) = 0$.)