1a) FALSE: Strassen's method is $\Theta(n^{\log_2 7})$

b) TRUE: each man is allowed to propose to each of *n* women.

or FALSE: possible to show that in worst case only n^2-n+1 proposals will be made.

c) FALSE: both are $\Theta(n^{2014})$

2a) $7n^2 - 3n \le 7n^2$ for all $n \ge 0$ so $O(n^2)$, c=7, $n_0=0$

$7n^2 - 3n \ge 7n^2 - 3n^2 = 4n^2$ for all $n \ge 0$ so $\Omega(n^2)$, c=4, $n_0=0$

b) $27n^{\sqrt{5}} \approx 27n^{2.236}$

$42(1/\sqrt{n})7^{\log_2 n} = 42n^{-0.5}n^{\log_2 7} \approx 42n^{2.807 - 0.5} = 42n^{2.307}$

$\Theta(n^{2.307})$ or $\Theta(n^{\log_2 7 - 0.5})$

3a) a = 5, b = 6: $x = \log_6 5 < 1$ (highest power of f(n))

Case 1: $\Theta(n)$

b) Initial assignment statements run in $\Theta(1)$ time.

One execution of the body of the while loop runs in $\Theta(1)$ time.

Complexity is then based on the number of executions of the while loop.

At end of iteration i of the while loop, we have $j = i^2$ and $k = 2i + 1$.

Number of iterations of while loop is $\lceil\sqrt{n}\rceil$ so $\Theta(\sqrt{n})$.

4. Basis. $n=1$: $T(1) = 1 \le cn = c$ so true for all $c \ge 1$.

Assume $T(k) \le c(k)$ for $1 \le k \le n-1$. Prove for $k = n$.

Let $\Theta(1)$ be some constant amount of work d.

$$\mathrm{T}(n) = \mathrm{T}\left(\frac{n}{3}\right) + \mathrm{T}\left(\frac{n}{6}\right) + \mathrm{T}\left(\frac{7n}{18}\right) + \mathrm{d}$$

$$\le c\left(\frac{n}{3}\right) + c\left(\frac{n}{6}\right) + c\left(\frac{7n}{18}\right) + \mathrm{d}$$

$$= c\left(\frac{16n}{18}\right) + \mathrm{d}$$

$\le cn + \mathrm{d}$ for all $n \ge 0$ since $16/18 < 1$

Out by a constant d.

- Asymptotic notation would simply drop drop the constant term here.
- Definition of big-O: we could choose $n_0 > d$ and $c \ge 1$.
- We could redo the induction and instead prove something stronger such as $T(n) \le cn - d$.

5a) Note: $n = 5^k$

Level 0 has 1 node of $5^k/\log 5^k$

Level 1 has 5 nodes of $5^{k-1}/\log 5^{k-1}$

Level 2 has 25 nodes of $5^{k-2}/\log 5^{k-2}$

b) n or 5^k

c) k+1

d) Assume log base 5. Level i work sums to $5^k/\log 5^{k-i}$

$$3(5^k) + \sum_{i=0}^{k-1} 5\mathrm{k}/\log 5^{\mathrm{k-i}} = 3n + n\sum_{i=0}^{k-1}\frac{1}{log5^{k-i}} = 3n + \frac{n}{log5}\sum_{i=0}^{k-1}\frac{1}{k-i} = 3n + n\sum_{i=1}^{k}\frac{1}{i}$$

$\in \Theta(n + n\log k) = \Theta(n\log\log n)$

6a) Sort and relabel customer locations so that $c_1 \le c_2 \le \ldots \le c_n$.

```
x₁ = c₁ + D
i = 1
j = 2
while j ≤ n
   if cⱼ is in range of xᵢ // check if |xᵢ - cⱼ| ≤ D
      j++
   else
      i++
      xᵢ = cⱼ + D
      j++
```

b) Exchange Proof. Since **O** is an optimal solution, $p \le k$.

Claim: $o_1 \le g_1$, $o_2 \le g_2$, ..., $o_k \le g_k$.

Proof by math induction. Assume the first ℓ greedy choices can be the start of an optimal solution.

i.e. $\{ g_1, \ldots, g_\ell, o_{\ell+1}, \ldots, o_p \}$ is an optimal solution for all ℓ upto k.

Basis, $\ell = 0$: $\{ o_1, \ldots, o_p \}$ is an optimal solution.

Assume $\{ g_1, \ldots, g_{\ell-1}, o_\ell, \ldots, o_p \}$ is an optimal solution.

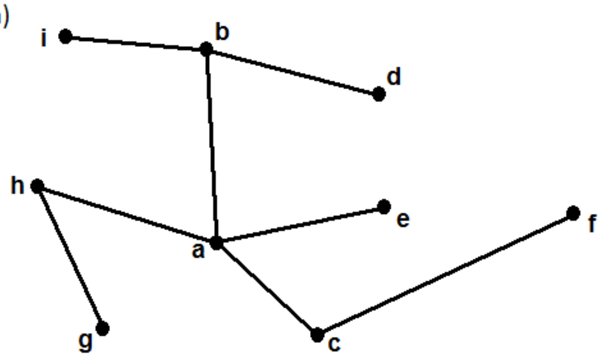
Prove $\{ g_1, \ldots, g_{\ell-1}, g_\ell, o_{\ell+1}, \ldots, o_p \}$ is an optimal solution; i.e. we can replace o_ℓ with g_ℓ for all ℓ upto k and it will remain an optimal solution.

The first ℓ-1 locations match the greedy solution. Next, the greedy algorithm finds the first customer out of range of the first ℓ-1 locations and places a new store at this location + D (i.e. it places a store as far away as possible but making sure all customers have a store they can get to) so $g_\ell \ge o_\ell$ and $\{ g_1, \ldots, g_\ell \}$ covers all of the customers (and possibly more than) $\{ g_1, \ldots, g_{\ell-1}, o_\ell \}$. Thus, $\{ g_1, \ldots, g_k, o_{k+1}, \ldots, o_p \}$ is an optimal solution.

Also, k = p since if after the kth choice there were still customers without a store in range, the greedy algorithm would continue to make choices.

7.

a)



b)