Homework 1

Simon Zheng
260744353

September 22nd, 2017

1 Stable marriage matching problem

a)

$$m_1 : w_1, w_2, w_3, w_4$$
$$m_2 : w_2, w_1, w_3, w_4$$
$$m_3 : w_3, w_2, w_1, w_4$$
$$m_4 : w_4, w_2, w_3, w_1$$
$$w_1 : w_4, w_2, w_3, w_1$$
$$w_2 : w_1, w_4, w_3, w_2$$
$$w_3 : w_1, w_2, w_4, w_3$$
$$w_4 : w_1, w_2, w_3, w_4$$

Simply assume each man has a different woman as first choice (and they are respectively the worst choice of the woman). Since the algorithm is ran on the men, each proposes to their first choice one after the other and the women must accept. The men also do not break up any pairings as they are all different. The algorithm terminates, so the rest doesn't even matter.

b) .

c)

$$m_1 : m_2, m_3, m_4$$
$$m_2 : m_3, m_4, m_1$$
$$m_3 : m_4, m_2, m_1$$
$$m_4 : m_1, m_2, m_3$$

While the algorithm terminates, it is not a stable matching.

$$
\begin{aligned}
m_1 &: \cancel{m_2}, \underline{m_3}, \cancel{m_4} \\
m_2 &: \cancel{m_3}, \underline{m_4}, \cancel{m_1} \\
m_3 &: \cancel{m_4}, \cancel{m_2}, \underline{m_1} \\
m_4 &: \cancel{m_1}, \underline{m_2}, \cancel{m_3}
\end{aligned}
$$

2

a)

$$
\begin{aligned}
\sqrt{n} + n\sqrt{n} &= \sqrt{n}(1+n) \\
&\leq n(1+n) \\
&\leq n(n+n) \\
&= n^2 + n^2 \\
&= 2n
\end{aligned}
$$

Thus, for $c = 2$ and $n_0 = 3$.

b) On one hand,

$$
\begin{aligned}
(n + \log_2 n)^5 &\geq n^5 + (\log_2 n)^5 \geq n^5 \\
n^5 &= O((n + \log_2 n)^5)
\end{aligned}
$$

For $c = 1, n_0 = 1$. On the other hand,

$$
\begin{aligned}
(n + \log_2 n)^5 &\leq (n+n)^5 = 2^5 n^5 \\
(n + \log_2 n)^5 &= O(n^5)
\end{aligned}
$$

For $c = 2^5, n_0 = 1$.

3

a)

$$
\begin{aligned}
n! &= 1 \times 2 \times ... \times (n-1) \times n \\
n^n &= n \times n \times ...n\ times... \times n \times n
\end{aligned}
$$

So,

$$
\begin{aligned}
\lim_{n\to\infty} \frac{n!}{n^n} &= \lim_{n\to\infty} \frac{1 \times 2 \times ... \times (n-1) \times n}{n \times n \times ... \times n \qquad \times n} \\
&= 0
\end{aligned}
$$

Therefore, $n! = o(n^n)$.

b) .

4

a) Big-Oh of a sum of function is the same as big-Oh of the fastest growing function. We can see $O(f(n))$ as the set of all functions that are upper-bounded by $f(n)$. $O(f(n)+g(n)) = O(max(f(n), g(n)))$ So on one hand if $f(n) > g(n)$ (as in grows faster) then

$$\begin{aligned} O(f(n)+g(n)) &= O(f(n)) \\ O(f(n)) &= f(n) + O(g(n)) \\ &= f(n) \end{aligned}$$

as $f(n) \geq g(n)$ so any function that is $O(g(n))$ will still be dominated by the faster growing $f(n)$. On the other hand, if $g(n)$ dominates over $f(n)$, then

$$\begin{aligned} O(f(n)+g(n)) &= O(g(n)) \\ O(g(n)) &= f(n) + O(g(n)) \\ &= O(g(n)) \end{aligned}$$

as since $g(n)$ grows faster than $f(n)$, there will be at least one function $O(g(n))$ that exists that grows faster than $f(n)$ thus $f(n)$ doesn't matter and is dominated by the set of functions $O(g(n))$.

b) .

Algorithm .1: f()

```
1 Input:
2 Output:
```