Homework 2

Nathaniel Graham

October 4, 2019

Problem 1

Exercise 2.1.4:

A) The glove selection
In the best case scenario, after selecting only two gloves we can have a pair.
The worst case scenario depends on the definition of a "pair"

Case 1: The glove pairs aren't left/right specific, any two gloves of the same color count as a pair.
In this case, the worst case scenario would involve drawing one glove of each color, which would be 3 gloves. The 4th glove would then create a guaranteed pair.

Case 2: The glove pairs are left/right specific, a right and a left glove that are of the same color are what constitutes a pair.
In this case, the worst case scenario is drawing all right or all left gloves (11 gloves), so the 12th glove would then guarantee that there was at least one matching pair of gloves.

B) Missing socks
With 5 distinct pairs of socks, lets look at the socks that could be taken after the first one is gone. After one sock is gone, any one of the nine other socks can be randomly taken as well. The chance that one sock is of the same pair as the first sock, is 1/9 (best case scenario), and the chance that it isn't, is 8/9 (worst case scenario). Therefore, 1/9 of the time, there are 4 pairs left, and 8/9 of the time, there are 3 pairs left. Combining these as follows, $4 * 1/9 + 3 * 8/9$, the result is 28/9. This is the average case.

Problem 2

Exercise 2.2.2:

a) $\frac{n(n+1)}{2} \in O(n^3)$
TRUE, because the degree of the polynomial is 2, so it's within $O(n^2)$, which is within $O(n^3)$

b) $\frac{n(n+1)}{2} \in O(n^2)$
TRUE, because the degree of the polynomial is 2, so it's within $O(n^2)$

c) $\frac{n(n+1)}{2} \in \Theta(n^3)$
FALSE, because no positive constant c can make the function "bookend" the function n^3

d) $\frac{n(n+1)}{2} \in \Omega(n)$
TRUE, because all forms of the function are at most as efficient as constant time

e) $\frac{n(n+1)}{2} \in O(n^2 log_2 n)$
TRUE, because the degree of the polynomial is 2, which is the same as the degree of $n^2 l_2 n$, so given a positive constant c, the polynomial can be at least as efficient as $n^2 log_2 n$

Exercise 2.2.3:

a) $(n^2+1)^{10}$
$\Theta(n^{20})$, because the degree of the polynomial, ignoring the insignificant +1, is 20

b) $\sqrt{10n^2+7n+3}$
$\Theta(n)$, because the degree of the polynomial is n from $\sqrt{n^2}$

c) $2nlog(n+2)^2+(n+2)^2log(\frac{n}{2})$
$\Theta(n^2)$, because the degree of the polynomial is n^2, from $(n+2)^2$

d) $2^{n+1}+3^{n-1}$
($\Theta(3^n)$, since $3^{n-1}=\frac{1}{3}*3^n$ and therefore the degree of the polynomial is 3^n.

e) $\lfloor log_2 n \rfloor$
$\Theta(log_2 n), because flooring the equation doesn't affect its efficiency class$

Problem 3

Exercise 2.3.1:
c.

$$\sum_{i=3}^{n+1} 1 = n-1$$

d.

$$\sum_{i=3}^{n+1} i = \frac{n^2+3n-4}{2}$$

e.

$$\sum_{i=0}^{n-1} i(i+1) = \frac{n(n^2-1)}{3}$$

f.

$$\sum_{j=1}^{n} 3^{j+1} = \frac{3^{n+2}-9}{2}$$

g.

$$\sum_{i=1}^{n}\sum_{j=1}^{n} ij = \frac{n^2(n+1)^2}{4}$$

Problem 4

//input: nonnegative integer n

S = 0

```
FOR i = 1 to n do
. . S = S + i*i
. . return S
ENDFOR
```

a) What does this algorithm compute?
This algorithm computes $\sum_{i=1}^{n} i^2$

b) What is its basic operation?
The basic operation is multiplication.
c) How many times is the operation executed?
n times
d) What is the efficiency class of the algorithm?
O(n)
e) By using the polynomial equivalent equation $\frac{n(n+1)(n+2)}{6}$ then the efficiency class drops to O(1) or constant time, which is more efficient.

Problem 5

i)

```python
def main():
    the_input = input("Enter a number\n>>")
    the_primes = find_primes_less_than(int(the_input))
    print(the_primes)


def find_primes_less_than(n):
    the_range = list(range(2, n))
    for i in the_range:
        for j in the_range:
            if j != i and j % i == 0:
                the_range.remove(j)
    return the_range


if __name__ == "__main__":
    main()
```

ii) The way this algorithm works is by, starting from 2, removing all multiples of 2 from the range, then checking the next existing number and removing all multiples of it. Do this until all existing numbers have been checked (we've reached the last number in the range). At this point, all multiples of any number have been removed; therefore, there are only primes left in the range.

iii) [2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163, 167, 173, 179, 181, 191, 193, 197, 199, 211, 223, 227, 229, 233, 239, 241, 251, 257, 263, 269, 271, 277, 281, 283, 293, 307, 311, 313, 317, 331, 337, 347, 349, 353, 359, 367, 373, 379, 383, 389, 397, 401, 409, 419, 421, 431, 433, 439, 443, 449, 457, 461, 463, 467, 479, 487, 491, 499]