Building Blocks Homework 12
Solution

Question A) For each of the following functions T(n) compute T(20), T(40) and the ratio T(40)/T(20).

- **a)** $T(n) = kn$
- **b)** $T(n) = kn^2$
- **c)** $T(n) = kn^3$
- **d)** $T(n) = k2^n$
- **e)** $T(n) = klog(n)$

a) T(20)=20k
T(40)=40k
T(40)/T(20) = 40k/20k = 2 (linear growth)

b) T(20)= $k20^2$=400k
T(40)= $k40^2$=1600k
T(40)/T(20) = 1600k/400k = 4 (quadratic growth)

c) T(20)= $k20^3$=8000k
T(40)= $k40^3$=64000k
T(40)/T(20) = 64000k/8000k = 8 (cubic growth)

d) T(20)= $k2^{20}$=1048576k
T(40)= $k2^{40}$= 1,099,511,627,776k
T(40)/T(20) = 1048576 (exponential growth)

e) T(20)= $klog(20)$ ≈ 1.3k
T(40)= $klog(40)$ ≈ 1.6k
T(20)/T(10) = 1.6k/1.3k ≈ 1.23k (logarithmic growth)

Question B) For each of the following pairs of functions determine

1) Which function is smaller when x is positive and close to zero.
2) Which function is smaller when x is positive and very large.
3) Find any cross over points for the two functions. Estimate the value if a direct solution is not possible.

a) $f(x) = 10x$ and $g(x) = 5x^2$

b) $f(x) = x^3$ and $g(x) = 3^x$ (Estimate crossover)

c) $f(x) = 10000000$ and $g(x) = log(log(x))$

a) When x is small g(x) is smaller. When x is large f(x) is smaller. We set f(x)=g(x) and solve to find the crossover.

$$5x^2 = 10\mathrm{x} \qquad \text{move terms}$$
$$5x^2 - 10x = 0 \qquad \text{and factor}$$
$$\mathrm{x}(5\mathrm{x} - 10) = 0$$

We get two solutions, x=0 and x=2

b) When x is small g(x) is smaller. When x is large f(x) is smaller. We set f(x)=g(x) and solve to find the crossover.

$$x^3 = 3^{\mathrm{x}} \qquad \text{or after taking a log of both sides}$$
$$\log(\mathrm{x}^3) = \log(3^x) \qquad \text{or}$$
$$3\log(\mathrm{x}) = \mathrm{x}\log(3) \qquad \text{or}$$
$$\mathrm{x} = \frac{3\log(\mathrm{x})}{\log(3)}$$

Graphing, we see that this has two solutions at approximately x=2.49 and x=2.98

c) When x is small g(x) is smaller. When x is large f(x) is smaller. We set f(x)=g(x) and solve to find the crossover.

$$log(log(x)) = 10000000 \qquad \text{exponentiate with a base of 10}$$
$$log(x) = 10^{10000000} \qquad \text{exponentiate again}$$
$$log(x) = 10^{10^{10000000}}$$

Loglog is an extremely slow growth function. It is basically between 1 and 2 for all reasonable values of x.

Question C) Prove that n^2 is $O(n^3)$. (Upper bounded)

Witnesses: C=10, k=5.
Show: $0 < n^2 \leq 10n^3 \; for \; all \; n \geq 5$

Since $n \geq 5$, we know that $n^2 \geq 5^2$ and $n^3 \geq 5^3$

This means that all of the terms are positive and we satisfy the zero condition and we are left showing

$$n^2 \leq 10n^3 \;\; for \;\; all \;\; n \geq 5$$

Since n squared is positive we can safely divide both sides

$$1 \leq 10\mathrm{n} \;\; for \;\; all \;\; n \geq 5$$

$$\mathrm{n} \geq 1/10 \;\; for \;\; all \;\; n \geq 5$$

We know that $n \geq 5$ and $5 \geq 1/10$ so we can conclude $n \geq 1/10$

Question D) Prove that n^5 is $\Omega(n^2)$. (Lower bounded)

Witnesses: C=2, k=5.
Show: $0 < 2n^2 \leq n^5 \; for \; all \; n \geq 5$

Since $n \geq 5$, we know that $n^2 \geq 5^2$ and $n^5 \geq 5^5$

This means that all of the terms are positive and we satisfy the zero condition and we are left showing

$$2n^2 \leq n^5 \; for \; all \; n \geq 5$$

Since n squared is positive we can safely divide both sides

$$2 \leq n^3 \; for \; all \; n \geq 5$$

$$n^3 \geq 2 \;\; for \;\; all \;\; n \geq 5$$

We know that $n \geq 5$ which means that $n^3 \geq 5^3$ combined with 5^3 $\geq$ 2 we can conclude $n^3 \geq 2$

Question E) Prove that $3n^3 + 5n^2 + 12$ is $\Theta(n^3)$.   (Both bounds)

Upper Bound
Witnesses: C=20 k=5   (I chose 20 because 3+5+12=20)
Show:

$0 \leq 3n^3 + 5n^2 + 12 \leq 20n^3$ for all n≥5

We know that n>5, so $n^2 \geq 5^2$ and $n^3 \geq 5^3$. All terms are positive and we can drop the zero.

$3n^3 + 5n^2 + 12 \leq 20n^3$ for all n≥5

We break this up into 3 pieces we need to show. If so we can combine to get the statement above.

1) $3n^3 \leq 3n^3$ for all n≥5   This is clearly true.
2) $5n^2 \leq 5n^3$ for all n≥5   Divide both sides by n^2 and we have
   $1 \leq$ n for all n≥5   Is true
3) $12 \leq 12n^3$ for all n>5   Divide by 12
   $1 \leq n^3$ for all n>5   Is true

Lower Bound:
Witnesses: C=2 k=5   (I can use any C less than 3)
Show:

$0 \leq 2n^3 \leq 3n^3 + 5n^2 + 12$ for all n≥5

We know that n>5, so $n^2 \geq 5^2$ and $n^3 \geq 5^3$. All terms are positive and we can drop the zero.

$2n^3 \leq 3n^3 + 5n^2 + 12$ for all n≥5

We break this up into 2 pieces we need to show. If so we can combine to get the statement above.

1) $2n^3 \leq 3n^3$ for all n≥5   Divide both sides by n^3.
   $2 \leq 3$ for all n≥5   Is clearly true
2) $0 \leq 5n^2 + 12$ for all n≥5   This is also clearly true since both terms on RHS are positive.

Question F) Consider the following algorithm that takes a list of values of size n and determines if there are three matching values in the list. Give a good Big-O estimate for the time required .

Algorithm: Determine if there are three matching values in the list.

```
def three_match(list):
  found = false         O(1)
  partial1 = false      O(1)
  partial2 = false      O(1)
  n = len(list)         O(1)
  # range(a,b) provides integer values from a to b-1
  for i in range(0, n):          O(n) times in loop
       for j in range(0, n):     O(n) times in loop
            partial1 = i!=j and list[i] == list[j]  O(1)
            for k in range(0, n):  O(n) times in loop
                 partial2 = j!=k and list[j] == list[k] O(1)
                 if partial1 and partial2 and i!=k:
                      found = true                  O(1)
  return found
```

The inner loop on k does O(n) work.
The loop on j does n times O(n) work, which is O(n^2)
The outer loop on I does n times O(n^2) work, which is O(n^3)
The initial O(1) overhead is lower order and ignored.

Final is O(n^3)

Set of n items Select r items	Order matters Permutation "Arrange"	Order does not matter Combination "Choose"
No repetition	$P(n,r) = \frac{n!}{(n-r)!}$	$C(n,r) = \frac{n!}{(n-r)!\,r!}$
Repetition allowed	n^r	$C(n-1+r,r)$

Question G) Suppose that we are going to build a sundae where we can select from 31 flavors of ice cream and 10 kinds of toppings. In all the following cases, you can select 4 flavors of ice cream and 1 kind of topping.

- **a)** What is the number of sundaes you can create provided that the scoops of ice cream are stacked on top of each other and each flavor must be different?
- **b)** What is the number of sundaes you can create provided that the scoops of ice cream are stacked on top of each other and you may repeat flavors?
- **c)** What is the number of sundaes you can create provided that the scoops of ice cream are swirled together and each flavor must be different?
- **d)** What is the number of sundaes you can create provided that the scoops of ice cream are swirled together and you may repeat flavors?

Hint: Use two tasks. Options a and b are ordered

First task

a) n=31, r=4, order matters so this is a permutation without repetitions.

$$P(31,4) = \frac{31!}{(31-4)!} = \frac{31!}{(27)!} = 31 \times 30 \times 29 \times 28$$

b) n=31, r=4, order matters so this is a permutation with repetitions.

$$31^4$$

c) n=31, r=4, order does not matter so this is a combination without repetitions.

$$C(31,4) = \frac{31!}{(31-4)!\times 4!} = \frac{31!}{(27)!\times 4!} = \frac{31\times 30\times 29\times 28}{4\times 3\times 2\times 1} = 31 \times 5 \times 29 \times 7$$

d) n=31, r=4, order does not matter so this is a combination with repetitions.

$$C(n+r-1,r)=C(34,4)$$

$$= \frac{34!}{(34-4)!\times 4!} = \frac{34!}{(30)!\times 4!} = \frac{34\times 33\times 32\times 31}{4\times 3\times 2\times 1} = 34 \times 11 \times 4 \times 31$$

The second task is to choose 1 topping from 10 and there 10 ways to do that. Final answers are obtained by multiplying to get

a) $31 \times 30 \times 29 \times 28 \times 10$
b) $31^4 \times 10$
c) $31 \times 5 \times 29 \times 7 \times 10$
d) $34 \times 11 \times 4 \times 31 \times 10$

Question H) There are 20 students in class. We will select 3 students in the class to receive a gift card and 2 students to get a tee-shirt. A student can only receive one item. How many ways are there to choose the students that receive prizes? (Hint: Use tasks that are combinations.)

Task 1: Choose 3 students to get the gift card:

Ways $C(20,3) = \frac{20!}{(17)!\times 3!} = \frac{20\times 19\times 18}{3\times 2\times 1} = 20 \times 19 \times 3$

Task 2: Choose 2 students for t-shirt. The set to select from is smaller by 3.

Ways $C(17,2) = \frac{17!}{(15)!\times 2!} = \frac{17\times 16}{2\times 1} = 17 \times 8$

Total is: $20 \times 19 \times 3 \times 17 \times 8$

Question I) The PawnGo Casino offers a game of chance that anyone can pay a dollar to play. You get a card with the numbers 1 to 80 on it. You mark 20 numbers (order doesn't matter) and win 100,000,000 dollars if you match a random drawing from those 80 numbers that the casino holds. How many ways can you mark the card?

There are 20 selections from 80, order does not matter, no repetitions. This is a standard combination.

$$C(80,20) = \frac{80!}{(60)! \times 20!} = \frac{80 \times 79 \times 78 \ldots \times 62 \times 61}{20 \times 19 \times 18 \ldots \times 2 \times 1} = 3{,}535{,}316{,}142{,}212{,}174{,}320$$

Your chance of getting the 100 million payout is 1 in C(80,20). The chance someone gets this is worse than the big lotteries which have jackpot odds of about 1 in 200,000,000 or so.

You may choose to solve one (and only one) of the following Extra Credit Problems. If you submit more than one, only the first will be graded.

Extra Credit 1)

Show that n^{10} is $O\left(\sqrt{3}^n\right)$.

We need witnesses: C=10^{1000} and k=5 (Big is good)

$$0 < n^{10} \leq 10^{1000}\sqrt{3}^n \quad \text{for all } n \geq 5$$

Since $n \geq 5$ all terms are positive

$$n^{10} \leq 10^{1000}\sqrt{3}^n \quad \text{for all } n \geq 5$$

We can take a log of both sides

$$10log(\mathrm{n}) \leq 1000 + \mathrm{nlog}(\sqrt{3}) \quad \text{for all } n \geq 5$$

Or

$$nlog(\sqrt{3}) \geq 10log(\mathrm{n}) - 1000$$

$$n \geq \frac{10(log(\mathrm{n}) - 100)}{log(\sqrt{3})} = 41.918065485787692 \times (log(\mathrm{n}) - 100)$$

Since n>5, log(n)>0.7. The inequality holds.

Extra Credit 2)

Give a good big O analysis of the running time of the following algorithm for best case and worst case.

Algorithm 2: Determine if there are three matching values in the list.

```
def three_match(list):
  n = len(list)
  # range(a,b) provides integer values from a to b-1
  for i in range(0, n):
       for j in range(i+1, n):
            for k in range(j+1, n):
                 if list[i] == list[j] and list[j] == list[k]:
                      return true
  return false
```

In this case, there is a return deep in the looping structure. If we get lucky, we find the match first time and the work is O(1). In the worst case, we run all the loops to completion. The loops are structured so that each combination of values from 1 to n are explored exactly once. The upperbound on the work is C(n,3) = O(n^3)