Final Exam Review

Day 2

Topics Covered

- Discrete Probability
- Conditional Probability
- Bayes' Theorem
- Expectation
- Complexity

Discrete Probability

Cheat sheet suggestions

Let E be an event in a sample space S. The probability of the event $\overline{E} = S - E$, the complementary event of E, is given by

$$p(\overline{E}) = 1 - p(E).$$

Let E_1 and E_2 be events in the sample space S. Then

$$p(E_1 \cup E_2) = p(E_1) + p(E_2) - p(E_1 \cap E_2).$$

Let E and F be events with $p(F) > 0$. The *conditional probability* of E given F, denoted by $p(E \mid F)$, is defined as

$$p(E \mid F) = \frac{p(E \cap F)}{p(F)}.$$

The probability of exactly k successes in n independent Bernoulli trials, with probability of success p and probability of failure $q = 1 - p$, is

$$C(n, k)p^k q^{n-k}.$$

$$p(E) = \frac{|E|}{|S|},$$

$$\sum_{s \in S} p(s) = 1.$$

The events E and F are *independent* if and only if $p(E \cap F) = p(E)p(F)$.

Discrete Probability - Terminology

Experiment - Procedure that yields an outcome

Sample Space - Set of all possible outcomes

Event - Subset of the sample space

If S is a set of equally likely outcomes, then the probability of an event E is:

$$p(E) = \frac{|E|}{|S|}$$

Probabilities

- Often combined with counting
- It can make the problem easier sometimes if you consider something as a permutation rather than combination (i.e. 1,6 is different than 6,1)
- Always start by calculating the sample space
- Disjoint does not mean Independent and vice versa!

Probability problems

- First define what is success and what is failure
- Divide the problem into cases, and then add them up later
- Make sure they are partitions
- Can you apply the binomial/geometric distribution?

Bernoulli Trials and the Binomial Distribution

Binomial Distribution: We call the probability distribution of the number of successes in a sequence of *n* Bernoulli trials the Binomial Distribution

- The **probability of exactly *k* successes** in *n* independent Bernoulli trials, with *p = Pr(success)* and *q = Pr(failure) = 1 - p*, is

$$Pr(numSuccesses = k) = C(n,k)\ (p^k)\ (q^{n-k})$$

- The **expected number of successes** in *n* independent Bernoulli trials where *p = Pr(success)*, is

$$E(numSuccesses) = np$$

The Geometric Distribution

***Examining number of trials to get a success**

- **Geometric distribution:** A random variable X has a geometric distribution with parameter p if

$$Pr(X = k) = (1 - p)^{k-1}p$$

- The **expected value of X** (ie the expected number of trials to get a success) is

$$E(X) = 1/p$$

Discrete Probability

Q: If you flip a coin 6 times, what is the probability of getting 2 or 3 tails?

Discrete Probability - Solution

Q: If you flip a coin 6 times, what is the probability of getting 2 or 3 tails?

Solution:

|Sample Space| = |S| = 2^6 = 64

|E| = (# sequences with 2 T) + (# sequences with 3 T)

= C(6, 2) + C(6, 3) = (6 * 5) / (2 * 1) + (6 * 5 * 4) / (3 * 2 * 1) = 15 + 20 = 35

P(E) = |E| / |S| = **35 / 64**

Discrete Probability

Q: In a certain lottery, there are 10 possible numbers to choose from. 2 distinct numbers are the winning numbers. To play the lottery, you pick n (distinct) numbers from the 10 possible numbers and only win if both of the winning numbers are included in your n picks. What is the least number of picks, n, such that your chances of winning are at least 50%? (Hint: consider |E| / |S|)

Discrete Probability - Solution 1

Solution 1:

$|S|$ = # of ways to choose n numbers out of 10 = C(10, n)

$|E|$ = # of ways to choose n numbers when both winning numbers must be included in our n picks = # ways to pick both winning numbers * # ways to choose remaining numbers (without considering order) = C(2, 2) * (10 - 2, n - 2) =

P(winning) = $|E|$ / $|S|$ = C(8, n - 2) / C(10, n)

$$P(\text{winning}) = \frac{|E|}{|S|} = \frac{\binom{8}{n-2}}{\binom{10}{n}} = \frac{\frac{8!}{(n-2)!(10-n)!}}{\frac{10!}{n!(10-n)!}}$$

$$= \frac{8!}{(n-2)!\cancel{(10-n)!}} \cdot \frac{n!\cancel{(10-n)!}}{10!}$$

$$= \frac{\cancel{8!}}{\cancel{(n-2)!}} \cdot \frac{n\cdot(n-1)\cdot\cancel{(n-2)!}}{10\cdot 9\cdot\cancel{8!}} = \frac{n(n-1)}{90}$$

$$\frac{n(n-1)}{90} \geq .5 \rightarrow n(n-1) \geq 45 \rightarrow \boxed{n=8}$$

Discrete Probability - Solution 2

Solution 2:

The possible number of winning pairs from all 10 numbers = $C(10, 2) = 10 * 9 / 2 = 45 = |S|$

The possible number of winning pairs in your n picks = $C(n, 2) = n * (n - 1) / 2 = |E|$

$P(\text{winning}) = |E| / |S| = (n * (n - 1) / 2) / 45 \geq .5$

$\rightarrow n * (n - 1) / 2 \geq 22.5 \rightarrow n * (n - 1) \geq 45.$

If $n = 7$, then $7 * (7 - 1) = 42 \leq 45$, but if $n = 8$, then $8 * (8 - 1) = 56 \geq 45$, therefore our answer is **n = 8**.

Complements

- Let E be an event in a sample space S. The probability of the event $\overline{E}$, the complementary event of E (i.e., a set of outcomes that E does not happen), is given by

$$p(\overline{E}) = 1 - p(E)$$

Complements

Q: When rolling three dice, what is the probability that the sum of the faces is less than 17?

Complements - Solution

Q: When rolling three dice, what is the probability that the sum of the faces is less than 17?

Solution: The probability of rolling less than 17 is equal to 1 - the probability of NOT rolling less than 17:

$$P(< 17) = 1 - P(\geq 17) = 1 - (P(17) + P(18))$$

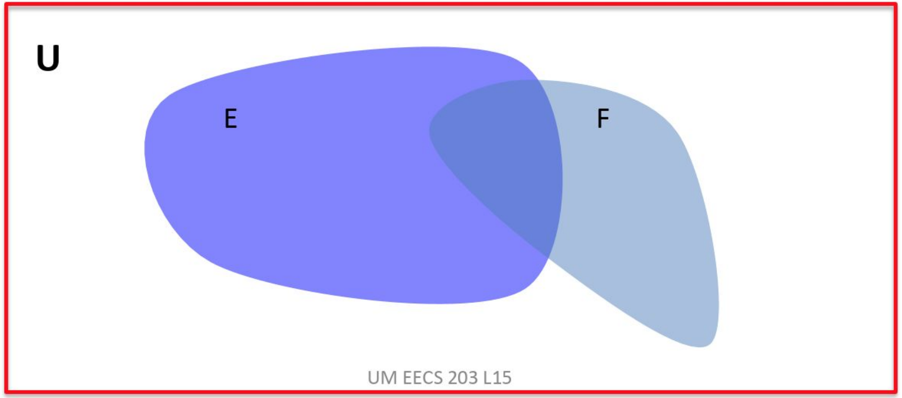
$$= 1 - (3 / 6^3 + 1 / 6^3) = (216 - 4) / 216 = \mathbf{53 / 54}$$

Conditional Probability

Conditional Probability

Let E and F be events with p(F) > 0. The **conditional probability** of E given F, defined by p(E | F), is defined as:

$$p(E \mid F) = \frac{p(E \cap F)}{p(F)}$$



Conditional Probability

Alice and Bob each get one chance to make a basketball shot. Alice has a probability of $\frac{4}{5}$ of making it, and Bob has a probability of $\frac{3}{5}$ of making it. Their probabilities are independent of each other. Given that at least one of them makes their shot, what is the probability that Alice makes hers?

(a) $\frac{23}{25}$

(b) $\frac{20}{23}$

(c) $\frac{4}{5}$

(d) $\frac{4}{7}$

(e) $\frac{12}{23}$

Conditional Probability Solution

Solution: b

Parse the situation first always:

P(A | AUB) = P(A∩(AUB)) / P(AUB) = P(A)/P(AUB)

[Don’t forget your sets just yet !!]

First use inclusive exclusive principle to calculate the probability that at least one of them makes their shot: P(A U B) = P(A) + P(B) - P(A)*P(B) = 23/25

P(A | AUB) = (⅘) / (23/25) = 4*25/(5*(23)) = **20/23**

Independence

The events E and F are **independent** if and only if
$p(E \cap F) = p(E)p(F)$

E and F are **independent** if and only if
$p(E \mid F) = p(E)$

Conceptually, independence means that knowing the outcome of E has no effect on the probability of F.

Independence

You roll a die 2 times.

- Let A be that the first roll is a 2
- Let B be that the sum of the two rolls is a 6.

Are A and B independent?

Independence - Solution

Question: You roll a die 2 times.

- Let A be that the first roll is a 2
- Let B be that the sum of the two rolls is a 6.

Solution: If the two events were independent, then $P(B \mid A) = P(B)$.

$P(B \mid A) = 1/6$ since the chance of getting a sum of 6 after rolling a 2 is equal to the chance of rolling a 4 on the second roll which is $1/6$.

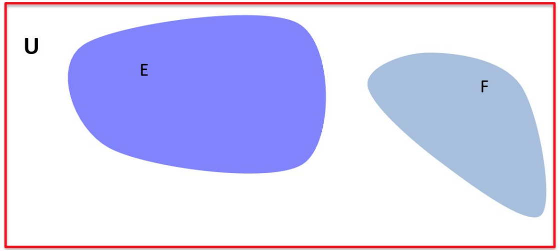
$P(B) = 5/36$

Since $1/6 \neq 5/36$, **A and B are not independent**.

Bayes' Theorem Cheat Sheet !!!

The probability of an event is the sum of the probabilities of all of the disjoint combinations of events that include it

$$p(E) = p(E \mid F)p(F) + p(E \mid \overline{F})p(\overline{F})$$



Suppose that E and F are events from a sample space S such that p(E) > 0 and p(F) > 0. Then

$$p(F \mid E) = \frac{p(E \mid F)p(F)}{p(E)}$$

$$p(F \mid E) = \frac{p(E \mid F)p(F)}{p(E \mid F)p(F) + p(E \mid \overline{F})p(\overline{F})}$$

When applying Bayes' Theorem to a situation that involves events Y and D (or any arbitrary notation), you can use the following identities

$$p(Y \mid D) + p(\overline{Y} \mid D) = 1$$
$$p(Y \mid \overline{D}) + p(\overline{Y} \mid \overline{D}) = 1$$
$$p(Y) + p(\overline{Y}) = 1$$
$$p(D) + p(\overline{D}) = 1$$

$$P(F_j \mid E) = \frac{P(E \mid F_j)p(F_j)}{\sum_{i=1}^{n} P(E \mid F_i)p(F_i)}$$

← Generalized Bayes'

Bayes' Theorem

A diagnostic test has a probability 0.9 of giving a positive result when used by someone with the disease and a probability of 0.2 of giving a false positive to those without the disease. 1 percent of the population have this disease. What is the probability

(a) that the result of the diagnostic test is positive

(b) that the test result and the actual state of the person do not agree

(c) that the person has the disease given that their results are negative

Solution:

Let T = tests positive and D = has the disease. $P(T|D) = 0.9$, $P(T|\bar{D}) = 0.2$, $P(D) = 0.01$

(a) $P(T) = P(T|D)P(D) + P(T|\bar{D})P(\bar{D}) = 0.9 \cdot 0.01 + 0.2 \cdot 0.99$

(b) $P(T \cap \bar{D}) + P(\bar{T} \cap D) = P(T|\bar{D})P(\bar{D}) + P(\bar{T}|D)P(D) = 0.2 \cdot 0.99 + 0.1 \cdot 0.01$

(c) $P(D|\bar{T}) = \frac{P(\bar{T}|D)P(D)}{P(\bar{T}|D)P(D)+P(\bar{T}|\bar{D})P(\bar{D})} = \frac{0.1 \cdot 0.01}{0.1 \cdot 0.01+0.8 \cdot 0.99}$

Expected Value

- Let p be a probability distribution over S.
- The ***expected value*** of X : S → ***R*** is the average of X over S, according to ("weighted by") the distribution p:

$$E(X) = \sum_{s \in S} p(s) \cdot X(s) = \sum_{r} p(X = r) \cdot r$$

This sum is over the ***outcomes*** in sample space S

This sum is over the possible ***values*** that X can take.

Linearity of Expectations

- The expected value of the sum of random variables is the sum of their expectations
- When we apply linear transformations to a random variable, we can take the scalars (a and b) outside of the expectation

$$E(X_1 + X_2) = E(X_1) + E(X_2)$$

$$E(a \cdot X + b) = a \cdot E(X) + b$$

1. Suppose we have a single answer multiple choice exam with 50 questions where each problem has 5 choices. You receive $+2$ points for correctly answering a problem and $-\frac{1}{2}$ points for incorrectly answering a problem. Assuming you guessed on all the problems on the exam, what grade would you expect?

(a) 0

(b) 5

(c) 10

(d) 6

(e) 20

Solution: a, let X = score on a single problem. $E(X) = \frac{1}{5} \cdot 2 + \frac{4}{5} \cdot (-\frac{1}{2}) = 0$. Then by linearity of expectations we have $50E(X) = 50 \cdot 0 = 0$

Consider the following function h, which takes a lowercase letter as input, and gives a positive integer between 1 and 26 as output:

$$h(\text{“}a\text{”}) = 1, h(\text{“}b\text{”}) = 2, h(\text{“}c\text{”}) = 3, \ldots, h(\text{“}z\text{”}) = 26$$

Another function f takes in a string of lowercase letters and gives the sum of the outputs of h for each character in the string. As an example:

$$f(\text{“}ace\text{”}) = 1 + 3 + 5 = 9$$

Consider a process which randomly chooses an integer n from 1 to 8 inclusive (each of equal probability), randomly generates a string s of length n (consisting of only lowercase letters, each of equal probability), and then computes $f(s)$. What is the expected value of this process?

Scantron 9

(A) $\frac{13 \cdot 14}{2}$

(B) 52

(C) $\frac{3^5}{4}$

(D) $\frac{13 \cdot 9}{4}$

(E) $\frac{13 \cdot 9}{2}$

Solution: C

The expected value of n is given by $E(X) = \sum_{i=1}^{8} i \cdot p(X = i) = \sum_{i=1}^{8} i \cdot \frac{1}{8} = \frac{36}{8} = \frac{9}{2}$. For any single character, the expected value of the value returned by h is given by $E(Y) = \sum_{j=1}^{26} j \cdot p(Y = j) = \sum_{j=1}^{26} j \cdot \frac{1}{26} = \frac{1}{26}\frac{26(26+1)}{2} = \frac{27}{2}$ (using $\sum_{k=1}^{n} k = \frac{n(n+1)}{2}$). As a result, the expected value of the value returned by f is given by $\frac{9}{2} \cdot \frac{27}{2} = \frac{3^2 \cdot 3^3}{4} = \frac{3^5}{4}$.

Break

Bernoulli and Binomial Distributions

- Bernoulli Trial
  - Success probability p, failure probability q = (1-p)

> The probability of exactly k successes in n independent (and identically distributed) Bernoulli trials is $\binom{n}{k} p^k q^{n-k}$

← referred to as a binomial distribution

A term in the binomial expansion of $(p+q)^n$

$$(p+q)^n = \sum_{k=0}^{n} \binom{n}{k} p^k q^{n-k} = \binom{n}{0} q^n + \binom{n}{1} p^1 q^{n-1} + \ldots + \binom{n}{n} p^n$$

- What is $(p+q)^n$?
  - $(p+(1-p))^n = (1)^n = 1$. Why is this important?
  - Answer: The sum of the probabilities must be 1.

Binomial

$E(X) = np$

Biased Coin Question

Suppose we have a biased coin, where the probability of heads is 0.6, and the probability of tails is 0.4. This coin is tossed 100 times.

Problem 15. (4 points)
What is the probability that there will be at most 99 heads?

`Scantron 15`

(A) $\sum_{k=1}^{99} \binom{100}{k} (0.6)^k (0.4)^{100-k}$

(B) $1 - (0.6)^{100}$

(C) All of the above

(D) None of the above

Biased Coin Solution

Solution: B

We could have solved this problem in two ways: the sum of probabilities of all cases that satisfy the condition “at most 99 heads”, or 1− (the sum of probabilities of cases that do not satisfy the condition “at most 99 heads”). In this case, only one case does not satisfy the condition (the case of 100 heads), thus the best solution is $1 - p(100 \text{ heads})$. The probability that we get 100 heads is $(0.6)^{100}$, thus answer choice B works. If we wanted to do it the other way, then answer choice A is correct except for the fact that it should start at $k = 0$ instead of $k = 1$.

Graph Expectation Problem

Given a complete graph K_n and colors red and blue, if all edges in the graph are individually colored red with probability q and blue with probability $1 - q$, what is the expected number of blue triangles in the resulting graph?

Solution: $\binom{n}{3}p^3$

We have a blue triangle with probability p^3, where $p = 1 - q$. There are C(n, 3) possible triangles in K_n, so by linearity of expectation, the expected number of triangles is C(n, 3) * p^3.

Complexity

Complexity Recap

- **Big-O:** A function $f(x)$ is $O(g(x))$ if there exists a $C > 0$ and k such that for all $x > k$, $|f(x)| \leq C|g(x)|$. In other words, the growth rate of $g(x)$ as x goes to infinity is an upper bound of the growth rate of $f(x)$ as x goes to infinity.

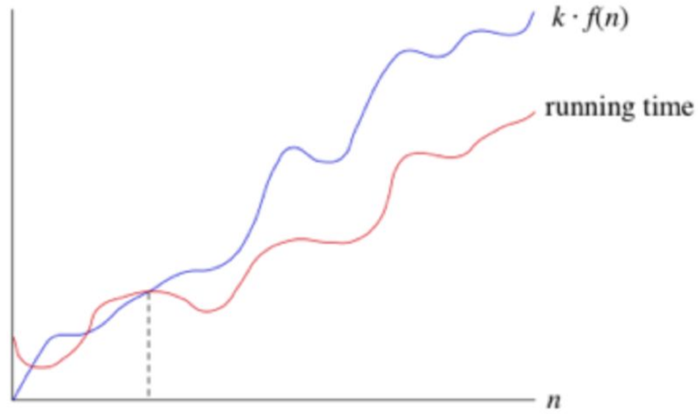
  f(x) is O(g(x)) -> f grows no faster than g; g is an upper bound for f

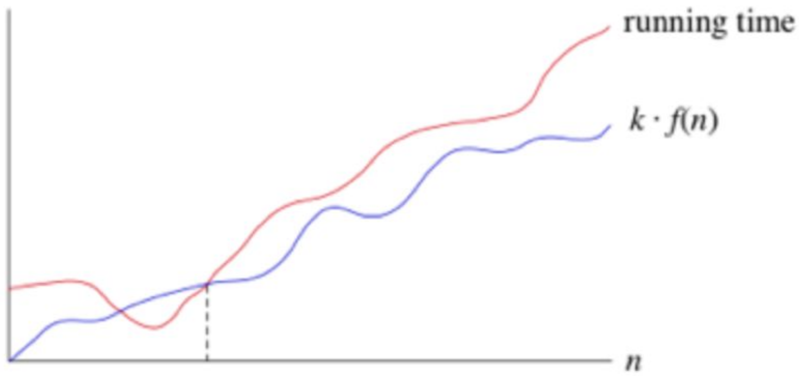
- **Big-Ω:** A function $f(x)$ is $\Omega(g(x))$ if there exists a $C > 0$ and k such that for all $x > k$, $C|g(x)| \leq |f(x)|$. In other words, the growth rate of $g(x)$ as x goes to infinity is a lower bound of the growth rate of $f(x)$ as x goes to infinity. This is the reverse of Big-O, so $f(x)$ is $\Omega(g(x))$ if and only if $g(x)$ is $O(f(x))$.

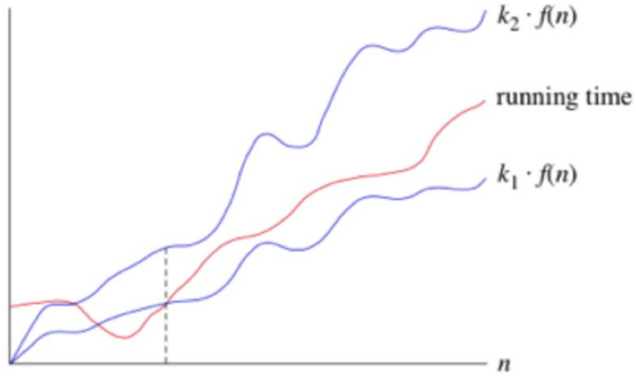
  f(x) is Ω(g(x)) -> f grows no slower than g; g is a lower bound for f

- **Big-Θ:** A function $f(x)$ is $\Theta(g(x))$ if $f(x)$ is $O(g(x))$ and $f(x)$ is $\Omega(g(x))$. This means that $f(x)$ and $g(x)$ grow at the same rate as x goes to infinity.

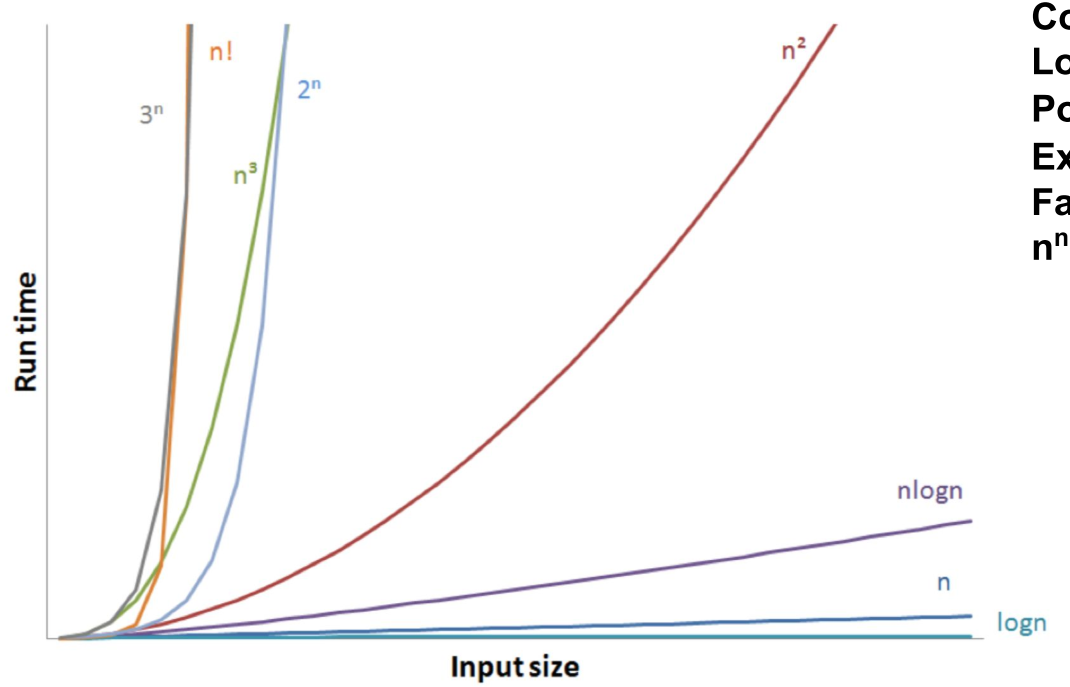
  f(x) is ϴ(g(x)) -> f and g grow at the same rate; f is both O(g(x)) and Ω(g(x))


$k \cdot f(n)$
running time
n


running time
$k \cdot f(n)$
n


$k_2 \cdot f(n)$
running time
$k_1 \cdot f(n)$
n

Cheat Sheet Suggestion



From low to high:

Constants (O(1))
Logarithmic (logn)
Polynomials (...$\sqrt{n}$, n, n^2, n^3,...)
Exponential (..., 2^n, 3^n,...)
Factorial (n!)
n^n

Log Properties

- $\log(xy) = \log(x) + \log(y)$
- $\log\left(\frac{x}{y}\right) = \log(x) - \log(y)$
- $\log(x^y) = y\log(x)$
- $\log_b(a) = \dfrac{\log_c(a)}{\log_c(b)}$ for all $a, b, c \in \mathbb{R}^+$ where $b > 1$ and $c > 1$
- $\log_b(b) = 1$ for all $b \in \mathbb{R}^+$ where $b > 1$

Simplifying Time Complexities

1. **Addition**: Highest complexity term dominates
   Q: What is the tightest big-O bound for $f(n) = 3^n + 4! + n^2$?
   A: $O(3^n)$

2. **Multiplication:** Multiply terms together for complexity
   Q: What is the tightest big-O bound for $f(x) = (x + \log(x))(x + 3)$?
   A: $O(x^2)$

3. **Scalar Addition and Multiplication:** Drop constants being added or multiplied
   Q: What is the tightest big-O bound for $f(n) = 203(3^n) + 400$
   A: $O(3^n)$

Problem: Time Complexity of Function

Which of the following are $O(2^n)$?

a. $f(n) = (\log n^3)^2 + n^2 + 2$
b. $f(n) = 3^n$
c. $f(n) = n^{203}$
d. $f(n) = n^2 \log n + 2^{\log n}$
e. $f(n) = 2^{(\log n)^2}$

Problem: Time Complexity of Function

Which of the following are $O(2^n)$?

a. $f(n) = (\log n^3)^2 + n^2 + 2$ — Yes; $9(\log n)^2 + n^2 + 2$; n^2 dominates -> $O(n^2)$ -> $O(2^n)$

b. $f(n) = 3^n$ — No; 3^n is higher complexity than 2^n

c. $f(n) = n^{203}$ — Yes; polynomials have lower complexity than exponentials

d. $f(n) = n^2 \log n + 2^{\log n}$ — Yes; complexity dominated by n ^2 log n. 2^log n = O(n), so the first term dominates, but it is still smaller than 2^n.

e. $f(n) = 2^{(\log n)^\wedge 2}$ — Yes; log n < all polynomials -> log n < $n^{1/2}$ (aka sqrt)

-> (log n)^2 < $(n^{1/2})^2$

-> (log n)^2 < n

-> $2^{(\log n)^\wedge 2} < 2^n$

Pseudocode Complexity Analysis Tips

- Look for three major things
  - Loops
  - Changes in indexes
  - Recursion
- Try running through one iteration yourself
- Multiply complexities for nested loops; add otherwise
- Remember that constants don't matter; only operations related to the size of the input matters

Problem: Pseudocode Analysis i

What is the runtime of the algorithm described by the following pseudocode?

function POWERSUM(N: integer)
  total := 0
  i := 1
  while $i \leq N$ **do**
    total := total + i
    $i := i * 2$
  end while
  return total
end function

(a) $\Theta(\log n)$

(b) $\Theta(\sqrt{n})$

(c) $\Theta(n)$

(d) $\Theta(n \log n)$

(e) $\Theta(2^n)$

Solution

What is the runtime of the algorithm described by the following pseudocode?

```
function PowerSum(N: integer)
    total := 0
    i := 1
    while i ≤ N do
        total := total + i
        i := i * 2
    end while
    return total
end function
```

i=2, 4, 8, 16… n -> log(n) times!

(a) $\Theta(\log n)$

(b) $\Theta(\sqrt{n})$

(c) $\Theta(n)$

(d) $\Theta(n \log n)$

(e) $\Theta(2^n)$

Problem: Pseudocode Analysis ii

Find the time complexity of the function

```
void function(int n)
{
    int count = 0;
    for (int i=n/2; i<=n; i++)
        for (int j=1; j<=n; j = 2 * j)
            for (int k=1; k<=n; k = k * 2)
                count++;
}
```

Solution

```
void function(int n)
{
    int count = 0;
    for (int i=n/2; i<=n; i++)
        for (int j=1; j<=n; j = 2 * j)
            for (int k=1; k<=n; k = k * 2)
                count++;
}
```

Runs n/2 times (O(n/2) = O(n))

Runs log(n) times (index is doubled in each iteration

$O(n \log^2 n)$

The outer loop repeats n/2 times (O(n)), and the two inner loops both repeat O(log n) times each as the index is multiplied by 2 every time. Combine complexity of nested loops via multiplication.

Problem: Pseudocode Analysis iii

Find the complexity of the following function

```
Procedure f(int n)
    i := 1
    While (i < n)
        i = i * 2
        For (1 to n)
            Print "good luck on the final!"
```

Solution

```
Procedure f(int n)

    i := 1

    While (i < n)

        i = i * 2

        For (1 to n)

            Print “good luck on the final!”
```

O(n log n)

The outer loop iterates log n times, and the inner loop iterates n times every time

The Master Theorem

Master Theorem: If the runtime for an algorithm can be modeled by a recurrence relation of the form $T(n) = aT\left(\frac{n}{b}\right) + \Theta(n^d)$ where $a > 0$, $b > 1$, and $d \geq 0$, then

$$T(n) \text{ is } \begin{cases} \Theta(n^d) & \text{if } \frac{a}{b^d} < 1 \\ \Theta(n^d \log n) & \text{if } \frac{a}{b^d} = 1 \\ \Theta(n^{\log_b a}) & \text{if } \frac{a}{b^d} > 1 \end{cases}$$

Master Theorem

- Used to analyze functions or pseudocode that is recursive
- The function must be in the correct form

$$T(n) = aT\left(\frac{n}{b}\right) + \Theta(n^d) \text{ where } a > 0,\ b > 1, \text{ and } d \geq 0,$$

  - At least one recursive call is made (a > 0)
  - The recursive call takes in a smaller, subproblem of the input (b > 1)
  - The remaining parts of the function take constant or polynomial time.

The Master Theorem: If $T(n) = aT(\frac{n}{b}) + n^d$, then

$$T(n) \text{ is } \begin{cases} \Theta(n^d) & \text{if } \frac{a}{b^d} < 1 \\ \Theta(n^d \log n) & \text{if } \frac{a}{b^d} = 1 \\ \Theta(n^{\log_b a}) & \text{if } \frac{a}{b^d} > 1 \end{cases}$$

Ex:

Let $T(n) = 2T\left(\frac{n}{4}\right) + \sqrt{n} + 42.$

The Master Theorem: If $T(n) = aT(\frac{n}{b}) + n^d$, then

$$T(n) \text{ is } \begin{cases} \Theta(n^d) & \text{if } \frac{a}{b^d} < 1 \\ \Theta(n^d \log n) & \text{if } \frac{a}{b^d} = 1 \\ \Theta(n^{\log_b a}) & \text{if } \frac{a}{b^d} > 1 \end{cases}$$

Solution:

Let $T(n) = 2T\left(\frac{n}{4}\right) + \sqrt{n} + 42.$

$$\begin{aligned} a &= 2 \\ b &= 4 \\ d &= \tfrac{1}{2} \end{aligned}$$

Therefore which condition?

Since $2 = 4^{\frac{1}{2}}$, case 2 applies.

Thus we conclude that

$$T(n) \in \Theta(n^d \log n) = \Theta(\sqrt{n} \log n)$$

Problems

Suppose we have an algorithm whose runtime $T(n)$ satisfies the recurrence relation

$$T(n) = 4T(n/2) + n^2.$$

Which of the following is the tightest Big-O complexity of the runtime of this algorithm?

(a) $O(n^2)$

(b) $O(n^2 \log n)$

(c) $O((\log n)^4)$

(d) $O(n^{\log_2(4)})$

(e) $O(n^4)$

The Master Theorem: If $T(n) = aT(\frac{n}{b}) + n^d$, then

$$T(n) \text{ is } \begin{cases} \Theta(n^d) & \text{if } \frac{a}{b^d} < 1 \\ \Theta(n^d \log n) & \text{if } \frac{a}{b^d} = 1 \\ \Theta(n^{\log_b a}) & \text{if } \frac{a}{b^d} > 1 \end{cases}$$

Solution

Suppose we have an algorithm whose runtime $T(n)$ satisfies the recurrence relation

$$T(n) = 4T(n/2) + n^2.$$

Which of the following is the tightest Big-O complexity of the runtime of this algorithm?

(a) $O(n^2)$

(b) $O(n^2 \log n)$

(c) $O((\log n)^4)$

(d) $O(n^{\log_2(4)})$

(e) $O(n^4)$

a=4, b=2, d=2
a/b^d = 4/2^2 = 4/4 = 1
Thus, T(n) is $\Theta(n^d \log n) = \Theta(n^2 \log n)$
Therefore, T(n) is also $O(n^2 \log n)$

The Master Theorem: If $T(n) = aT(\frac{n}{b}) + n^d$, then

$$T(n) \text{ is } \begin{cases} \Theta(n^d) & \text{if } \frac{a}{b^d} < 1 \\ \Theta(n^d \log n) & \text{if } \frac{a}{b^d} = 1 \\ \Theta(n^{\log_b a}) & \text{if } \frac{a}{b^d} > 1 \end{cases}$$

What is the tightest big-O bound on the runtime of this algorithm?

```
procedure something(n):
    for i := 1 to 7:
        something(n/3)
    m = 1
    for i := 1 to n:
        for j := 1 to n:
            for k := 1 to n:
                m = m + 3
    return
```

The Master Theorem: If $T(n) = aT(\frac{n}{b}) + n^d$, then

$$T(n) \text{ is } \begin{cases} \Theta(n^d) & \text{if } \frac{a}{b^d} < 1 \\ \Theta(n^d \log n) & \text{if } \frac{a}{b^d} = 1 \\ \Theta(n^{\log_b a}) & \text{if } \frac{a}{b^d} > 1 \end{cases}$$

What is the tightest big-O bound on the runtime of this algorithm?

```
procedure something(n):
    for i := 1 to 7:
        something(n/3)
    m = 1
    for i := 1 to n:
        for j := 1 to n:
            for k := 1 to n:
                m = m + 3
    return
```

Recursive calls on smaller inputs are being made; use Master Theorem

a=7 recursive calls being made via loop

b=3 each call is on input 3 times smaller

3 nested linear loops (O(n) each), so these loops are $O(n^3)$

All other loops/code is constant (O(1))

So non-recursive overhead (n^d) is n^3, so

d=3

The Master Theorem: If $T(n) = aT(\frac{n}{b}) + n^d$, then

$$T(n) \text{ is } \begin{cases} \Theta(n^d) & \text{if } \frac{a}{b^d} < 1 \\ \Theta(n^d \log n) & \text{if } \frac{a}{b^d} = 1 \\ \Theta(n^{\log_b a}) & \text{if } \frac{a}{b^d} > 1 \end{cases}$$

Solution

$a = 7$

$b = 3$

$d = 3$

$7/3^3<1$ so $O(n^3)$

Good luck on the final!