# CS 473G: Combinatorial Algorithms, Fall 2005
# Homework 0

Due Thursday, September 1, 2005, at the beginning of class (12:30pm CDT)

| | |
|---|---|
| Name: | |
| Net ID: | Alias: |

☐ I understand the Homework Instructions and FAQ.

---

- Neatly print your full name, your NetID, and an alias of your choice in the boxes above. Grades will be listed on the course web site by alias. Please write the same alias on every homework and exam! For privacy reasons, your alias should not resemble your name or NetID. By providing an alias, you agree to let us list your grades; if you do not provide an alias, your grades will not be listed. ***Never*** *give us your Social Security number!*

- Read the "Homework Instructions and FAQ" on the course web page, and then check the box above. This page describes what we expect in your homework solutions—start each numbered problem on a new sheet of paper, write your name and NetID on every page, don't turn in source code, analyze and prove everything, use good English and good logic, and so on—as well as policies on grading standards, regrading, and plagiarism. **See especially the course policies regarding the magic phrases "I don't know" and "and so on".** If you have *any* questions, post them to the course newsgroup or ask during lecture.

- Don't forget to submit this cover sheet with the rest of your homework solutions.

- This homework tests your familiarity with prerequisite material—big-Oh notation, elementary algorithms and data structures, recurrences, discrete probability, and most importantly, induction—to help you identify gaps in your knowledge. **You are responsible for filling those gaps on your own.** Chapters 1–10 of CLRS should be sufficient review, but you may also want consult your discrete mathematics and data structures textbooks.

- Every homework will have five required problems. Most homeworks will also include one extra-credit problem and several practice (no-credit) problems. Each numbered problem is worth 10 points.

---

| # | 1 | 2 | 3 | 4 | 5 | 6* | Total |
|---|---|---|---|---|---|---|---|
| Score | | | | | | | |
| Grader | | | | | | | |

1. Solve the following recurrences. State tight asymptotic bounds for each function in the form $\Theta(f(n))$ for some recognizable function $f(n)$. You do not need to turn in proofs (in fact, please *don't* turn in proofs), but you should do them anyway, just for practice. Assume reasonable but nontrivial base cases. **If your solution requires specific base cases, state them!**

   (a) $A(n) = 2A(n/4) + \sqrt{n}$

   (b) $B(n) = \max\limits_{n/3<k<2n/3} \big(B(k) + B(n-k) + n\big)$

   (c) $C(n) = 3C(n/3) + n/\lg n$

   (d) $D(n) = 3D(n-1) - 3D(n-2) + D(n-3)$

   (e) $E(n) = \dfrac{E(n-1)}{3E(n-2)}$ *[Hint: This is easy!]*

   (f) $F(n) = F(n-2) + 2/n$

   (g) $G(n) = 2G\big(\lceil (n+3)/4 \rceil - 5n/\sqrt{\lg n} + 6\lg\lg n\big) + 7\sqrt[8]{n-9} - \lg^{10} n/\lg\lg n + 11^{\lg^* n} - 12$

   $^\star$(h) $H(n) = 4H(n/2) - 4H(n/4) + 1$ *[Hint: Careful!]*

   (i) $I(n) = I(n/2) + I(n/4) + I(n/8) + I(n/12) + I(n/24) + n$

   $\bigstar$(j) $J(n) = 2\sqrt{n} \cdot J(\sqrt{n}) + n$
   *[Hint: First solve the secondary recurrence $j(n) = 1 + j(\sqrt{n})$.]*

2. Penn and Teller agree to play the following game. Penn shuffles a standard deck[1] of playing cards so that every permutation is equally likely. Then Teller draws cards from the deck, one at a time without replacement, until he draws the three of clubs ($3\clubsuit$), at which point the remaining undrawn cards instantly burst into flames and the game is over.

   The first time Teller draws a card from the deck, he gives it to Penn. From then on, until the game ends, whenever Teller draws a card whose value is smaller than the previous card he gave to Penn, he gives the new card to Penn. To make the rules unambiguous, they agree on the numerical values $A = 1$, $J = 11$, $Q = 12$, and $K = 13$.

   (a) What is the expected number of cards that Teller draws?

   (b) What is the expected *maximum* value among the cards Teller gives to Penn?

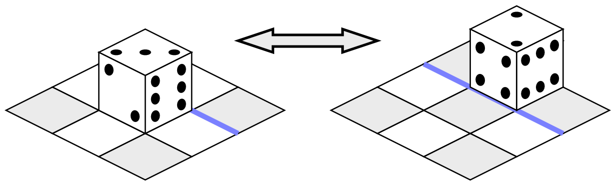
   (c) What is the expected *minimum* value among the cards Teller gives to Penn?

   (d) What is the expected number of cards that Teller gives to Penn?

   Full credit will be given only for *exact* answers (with correct proofs, of course).

[1]In a standard deck of 52 cards, each card has a *suit* in the set $\{\spadesuit, \heartsuit, \clubsuit, \diamondsuit\}$ and a *value* in the set $\{A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K\}$, and every possible suit-value pair appears in the deck exactly once. Penn and Teller normally use exploding razor-sharp ninja throwing cards for this trick.

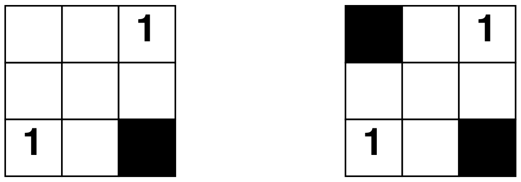
3. A *rolling die maze* is a puzzle involving a standard six-sided die[2] and a grid of squares. You should imagine the grid lying on top of a table; the die always rests on and exactly covers one square. In a single step, you can *roll* the die 90 degrees around one of its bottom edges, moving it to an adjacent square one step north, south, east, or west.



Rolling a die.

Some squares in the grid may be *blocked*; the die can never rest on a blocked square. Other squares may be *labeled* with a number; whenever the die rests on a labeled square, the number of pips on the *top* face of the die must equal the label. Squares that are neither labeled nor marked are *free*. You may not roll the die off the edges of the grid. A rolling die maze is *solvable* if it is possible to place a die on the lower left square and roll it to the upper right square under these constraints.

For example, here are two rolling die mazes. Black squares are blocked. The maze on the left can be solved by placing the die on the lower left square with 1 pip on the top face, and then rolling it north, then north, then east, then east. The maze on the right is not solvable.



Two rolling die mazes. Only the maze on the left is solvable.

(a) Suppose the input is a two-dimensional array $L[1..n][1..n]$, where each entry $L[i][j]$ stores the label of the square in the $i$th row and $j$th column, where 0 means the square is free and $-1$ means the square is blocked. Describe and analyze a polynomial-time algorithm to determine whether the given rolling die maze is solvable.

$^{\star}$(b) Now suppose the maze is specified *implicitly* by a list of labeled and blocked squares. Specifically, suppose the input consists of an integer $M$, specifying the height and width of the maze, and an array $S[1..n]$, where each entry $S[i]$ is a triple $(x, y, L)$ indicating that square $(x, y)$ has label $L$. As in the explicit encoding, label $-1$ indicates that the square is blocked; free squares are not listed in $S$ at all. Describe and analyze an efficient algorithm to determine whether the given rolling die maze is solvable. For full credit, the running time of your algorithm should be polynomial in the input size $n$.

*[Hint: You have some freedom in how to place the initial die. There are rolling die mazes that can only be solved if the initial position is chosen correctly.]*

[2]A standard die is a cube, where each side is labeled with a different number of dots, called *pips*, between 1 and 6. The labeling is chosen so that any pair of opposite sides has a total of 7 pips.

4. Whenever groups of pigeons gather, they instinctively establish a *pecking order.* For any pair of pigeons, one pigeon always pecks the other, driving it away from food or potential mates. The same pair of pigeons will always chooses the same pecking order, even after years of separation, no matter what other pigeons are around. (Like most things, revenge is a foreign concept to pigeons.) Surprisingly, the overall pecking order in a set of pigeons can contain cycles—for example, pigeon A pecks pigeon B, which pecks pigeon C, which pecks pigeon A.

   Prove that any set of pigeons can be arranged in a row so that every pigeon pecks the pigeon immediately to its right.

5. Scientists have recently discovered a planet, tentatively named "Ygdrasil", which is inhabited by a bizarre species called "vodes". All vodes trace their ancestry back to a particular vode named Rudy. Rudy is still quite alive, as is every one of his many descendants. Vodes reproduce asexually, like bees; each vode has exactly one parent (except Rudy, who has no parent). There are three different colors of vodes—cyan, magenta, and yellow. The color of each vode is correlated exactly with the number and colors of its children, as follows:

   - Each cyan vode has two children, exactly one of which is yellow.
   - Each yellow vode has exactly one child, which is not yellow.
   - Magenta vodes have no children.

   In each of the following problems, let $C$, $M$, and $Y$ respectively denote the number of cyan, magenta, and yellow vodes on Ygdrasil.

   (a) Prove that $M = C + 1$.

   (b) Prove that either $Y = C$ or $Y = M$.

   (c) Prove that $Y = M$ if and only if Rudy is yellow.

   *[Hint: Be very careful to* ***prove*** *that you have considered* ***all*** *possibilities.]*

$^\star$6. **[Extra credit]**[3]

*Lazy binary* is a variant of standard binary notation for representing natural numbers where we allow each "bit" to take on one of three values: 0, 1, or 2. Lazy binary notation is defined inductively as follows.

- The lazy binary representation of zero is 0.
- Given the lazy binary representation of any non-negative integer $n$, we can construct the lazy binary representation of $n + 1$ as follows:
  (a) increment the rightmost digit;
  (b) if any digit is equal to 2, replace the rightmost 2 with 0 and increment the digit immediately to its left.

Here are the first several natural numbers in lazy binary notation:

0, 1, 10, 11, 20, 101, 110, 111, 120, 201, 210, 1011, 1020, 1101, 1110, 1111, 1120, 1201, 1210, 2011, 2020, 2101, 2110, 10111, 10120, 10201, 10210, 11011, 11020, 11101, 11110, 11111, 11120, 11201, 11210, 12011, 12020, 12101, 12110, 20111, 20120, 20201, 20210, 21011, 21020, 21101, 21110, 101111, 101120, 101201, 101210, 102011, 102020, 102101, 102110, . . .

(a) Prove that in any lazy binary number, between any two 2s there is at least one 0, and between two 0s there is at least one 2.

(b) Prove that for any natural number $N$, the sum of the digits of the lazy binary representation of $N$ is exactly $\lfloor \lg(N + 1) \rfloor$.

[3]The "I don't know" rule does not apply to extra credit problems. There is no such thing as "partial extra credit".

## Practice Problems

The remaining problems are for practice only. Please do not submit solutions. On the other hand, feel free to discuss these problems in office hours or on the course newsgroup.

1. Sort the functions in each box from asymptotically smallest to asymptotically largest, indicating ties if there are any. You do not need to turn in proofs (in fact, please *don't* turn in proofs), but you should do them anyway, just for practice.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $1$ | $\lg n$ | $\lg^2 n$ | $\sqrt{n}$ | $n$ | $n^2$ | $2^{\sqrt{n}}$ | $\sqrt{2}^n$ |
| $2^{\sqrt{\lg n}}$ | $2^{\lg\sqrt{n}}$ | $\sqrt{2^{\lg n}}$ | $\sqrt{2}^{\lg n}$ | $\lg 2^{\sqrt{n}}$ | $\lg\sqrt{2}^n$ | $\lg\sqrt{2^n}$ | $\sqrt{\lg 2^n}$ |
| $\lg n^{\sqrt{2}}$ | $\lg\sqrt{n}^2$ | $\lg\sqrt{n^2}$ | $\sqrt{\lg n^2}$ | $\lg^2\sqrt{n}$ | $\lg^{\sqrt{2}} n$ | $\sqrt{\lg^2 n}$ | $\sqrt{\lg n}^2$ |

To simplify your answers, write $f(n) \ll g(n)$ to mean $f(n) = o(g(n))$, and write $f(n) \equiv g(n)$ to mean $f(n) = \Theta(g(n))$. For example, the functions $n^2, n, \binom{n}{2}, n^3$ could be sorted either as $n \ll n^2 \equiv \binom{n}{2} \ll n^3$ or as $n \ll \binom{n}{2} \equiv n^2 \ll n^3$.

2. Recall the standard recursive definition of the Fibonacci numbers: $F_0 = 0$, $F_1 = 1$, and $F_n = F_{n-1} + F_{n-2}$ for all $n \geq 2$. Prove the following identities for all positive integers $n$ and $m$.

   (a) $F_n$ is even if and only if $n$ is divisible by 3.

   (b) $\sum_{i=0}^{n} F_i = F_{n+2} - 1$

   (c) $F_n^2 - F_{n+1}F_{n-1} = (-1)^{n+1}$

   $^\star$(d) If $n$ is an integer multiple of $m$, then $F_n$ is an integer multiple of $F_m$.

3. Penn and Teller have a special deck of fifty-two cards, with no face cards and nothing but clubs—the ace, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, …, 52 of clubs. (They're big cards.) Penn shuffles the deck until each each of the 52! possible orderings of the cards is equally likely. He then takes cards one at a time from the top of the deck and gives them to Teller, stopping as soon as he gives Teller the three of clubs.

   (a) On average, how many cards does Penn give Teller?

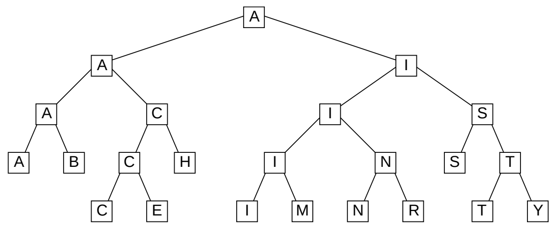
   (b) On average, what is the smallest-numbered card that Penn gives Teller?

   $^\star$(c) On average, what is the largest-numbered card that Penn gives Teller?

   Prove that your answers are correct. (If you have to appeal to "intuition" or "common sense", your answers are probably wrong.) *[Hint: Solve for an $n$-card deck, and then set $n$ to 52.]*

4. Algorithms and data structures were developed millions of years ago by the Martians, but not quite in the same way as the recent development here on Earth. Intelligent life evolved independently on Mars' two moons, Phobos and Deimos.[4] When the two races finally met on the surface of Mars, after thousands of years of separate philosophical, cultural, religious, and scientific development, their disagreements over the proper structure of binary search trees led to a bloody (or more accurately, ichorous) war, ultimately leading to the destruction of all Martian life.

   A *Phobian* binary search tree is a full binary tree that stores a set $X$ of search keys. The root of the tree stores the *smallest* element in $X$. If $X$ has more than one element, then the left subtree stores all the elements less than some pivot value $p$, and the right subtree stores everything else. Both subtrees are *nonempty* Phobian binary search trees. The actual pivot value $p$ is *never* stored in the tree.



A Phobian binary search tree for the set {M, A, R, T, I, N, B, Y, S, E, C, H}.

   (a) Describe and analyze an algorithm FIND$(x, T)$ that returns TRUE if $x$ is stored in the Phobian binary search tree $T$, and FALSE otherwise.

   (b) A *Deimoid* binary search tree is almost exactly the same as its Phobian counterpart, except that the *largest* element is stored at the root, and both subtrees are Deimoid binary search trees. Describe and analyze an algorithm to transform an $n$-node Phobian binary search tree into a Deimoid binary search tree in $O(n)$ time, using as little additional space as possible.

5. Tatami are rectangular mats used to tile floors in traditional Japanese houses. Exact dimensions of tatami mats vary from one region of Japan to the next, but they are always twice as long in one dimension than in the other. (In Tokyo, the standard size is 180cm×90cm.)

   (a) How many different ways are there to tile a $2 \times n$ rectangular room with $1 \times 2$ tatami mats? Set up a recurrence and derive an *exact* closed-form solution. *[Hint: The answer involves a familiar recursive sequence.]*

   (b) According to tradition, tatami mats are always arranged so that four corners never meet. How many different *traditional* ways are there to tile a $3 \times n$ rectangular room with $1 \times 2$ tatami mats? Set up a recurrence and derive an *exact* closed-form solution.

   $^\star$(c) How many different *traditional* ways are there to tile an $n \times n$ square with $1 \times 2$ tatami mats? Prove your answer is correct.

[4]Greek for "fear" and "panic", respectively. Doesn't that make you feel better?