

Change is certain. Peace is followed by disturbances; departure of evil men by their return. Such recurrences should not constitute occasions for sadness but realities for awareness, so that one may be happy in the interim.

— I Ching [The Book of Changes] (c. 1100 BC)

“... O Zarathustra, who you are and must become” behold you are the teacher of the eternal recurrence – that is your destiny! That you as the first must teach this doctrine – how could this great destiny not be your greatest danger and sickness too?

— Friedrich Nietzsche, *Also sprach Zarathustra* (1885)
[translated by Walter Kaufmann]

Wil Wheaton: Embrace the dark side!

Sheldon: That's not even from your franchise!

— “The Wheaton Recurrence”, *Bing Bang Theory*, April 12, 2010

Solving Recurrences

1 Introduction

A **recurrence** is a recursive description of a function, or in other words, a description of a function in terms of itself. Like all recursive structures, a recurrence consists of one or more *base cases* and one or more *recursive cases*. Each of these cases is an equation or inequality, with some function value $f(n)$ on the left side. The base cases give explicit values for a (typically finite, typically small) subset of the possible values of n . The recursive cases relate the function value $f(n)$ to function value $f(k)$ for one or more integers $k < n$; typically, each recursive case applies to an infinite number of possible values of n .

For example, the following recurrence (written in two different but standard ways) describes the identity function $f(n) = n$:

$$f(n) = \begin{cases} 0 & \text{if } n = 0 \\ f(n-1) + 1 & \text{otherwise} \end{cases} \quad \begin{aligned} f(0) &= 0 \\ f(n) &= f(n-1) + 1 \text{ for all } n > 0 \end{aligned}$$

In both presentations, the first line is the only base case, and the second line is the only recursive case. The same function can satisfy *many* different recurrences; for example, both of the following recurrences also describe the identity function:

$$f(n) = \begin{cases} 0 & \text{if } n = 0 \\ 1 & \text{if } n = 1 \\ f(\lfloor n/2 \rfloor) + f(\lceil n/2 \rceil) & \text{otherwise} \end{cases} \quad f(n) = \begin{cases} 0 & \text{if } n = 0 \\ 2 \cdot f(n/2) & \text{if } n \text{ is even and } n > 0 \\ f(n-1) + 1 & \text{if } n \text{ is odd} \end{cases}$$

We say that a particular function **satisfies** a recurrence, or is the **solution** to a recurrence, if each of the statements in the recurrence is true. Most recurrences—at least, those that we will encounter in this class—have a solution; moreover, if every case of the recurrence is an equation, that solution is unique. Specifically, if we transform the recursive formula into a recursive *algorithm*, the solution to the recurrence is the function computed by that algorithm!

Recurrences arise naturally in the analysis of algorithms, especially recursive algorithms. In many cases, we can express the running time of an algorithm as a recurrence, where the recursive cases of the recurrence correspond exactly to the recursive cases of the algorithm. Recurrences are also useful tools for solving counting problems—How many objects of a particular kind exist?

By itself, a recurrence is not a satisfying description of the running time of an algorithm or a bound on the number of widgets. Instead, we need a *closed-form* solution to the recurrence; this is a *non-recursive* description of a function that satisfies the recurrence. For recurrence *equations*, we sometimes prefer an *exact* closed-form solution, but such a solution may not exist, or may be too complex to be useful. Thus, for most recurrences, especially those arising in algorithm analysis, we can be satisfied with an *asymptotic* solution of the form $\Theta(f(n))$, for some explicit (non-recursive) function $g(n)$.

For recursive *inequalities*, we prefer a *tight* solution; this is a function that would still satisfy the recurrence if all the inequalities were replaced with the corresponding equations. Again, exactly tight solutions may not exist, or may be too complex to be useful, so we may have to settle for a looser solution and/or an asymptotic solution of the form $O(g(n))$ or $\Omega(g(n))$.

2 The Ultimate Method: Guess and Confirm

Ultimately, there is only one fail-safe method to solve *any* recurrence:

Guess the answer, and then prove it correct by induction.

Later sections of these notes describe techniques to generate guesses that are guaranteed to be correct, provided you use them correctly. But if you're faced with a recurrence that doesn't seem to fit any of these methods, or if you've forgotten how those techniques work, don't despair! If you guess a closed-form solution and then try to verify your guess inductively, usually either the proof will succeed, in which case you're done, or the proof will fail, in which case *the failure will help you refine your guess*. Where you get your initial guess is utterly irrelevant¹—from a classmate, from a textbook, on the web, from the answer to a different problem, scrawled on a bathroom wall in Siebel, included in a care package from your mom, dictated by the machine elves, whatever. If you can prove that the answer is correct, then it's correct!

2.1 Tower of Hanoi

The classical Tower of Hanoi problem gives us the recurrence $T(n) = 2T(n - 1) + 1$ with base case $T(0) = 0$. Just looking at the recurrence we can guess that $T(n)$ is something like 2^n . If we write out the first few values of $T(n)$, we discover that they are each one less than a power of two.

$$T(0) = 0, \quad T(1) = 1, \quad T(2) = 3, \quad T(3) = 7, \quad T(4) = 15, \quad T(5) = 31, \quad T(6) = 63, \quad \dots,$$

It looks like $T(n) = 2^n - 1$ might be the right answer. Let's check.

$$\begin{aligned} T(0) &= 0 = 2^0 - 1 & \checkmark \\ T(n) &= 2T(n - 1) + 1 \\ &= 2(2^{n-1} - 1) + 1 & [\text{induction hypothesis}] \\ &= 2^n - 1 & [\text{algebra}] \end{aligned}$$

We were right! Hooray, we're done!

¹... except of course during exams, where you aren't supposed to use *any* outside sources

Another way we can guess the solution is by *unrolling* the recurrence, by substituting it into itself:

$$\begin{aligned} T(n) &= 2T(n-1) + 1 \\ &= 2(2T(n-2) + 1) + 1 \\ &= 4T(n-2) + 3 \\ &= 4(2T(n-3) + 1) + 3 \\ &= 8T(n-3) + 7 \\ &= \dots \end{aligned}$$

It looks like unrolling the initial Hanoi recurrence k times, for any non-negative integer k , will give us the new recurrence $T(n) = 2^k T(n-k) + (2^k - 1)$. Let's prove this by induction:

$$\begin{aligned} T(n) &= 2T(n-1) + 1 \quad \checkmark & [k=0, \text{ by definition}] \\ T(n) &= 2^{k-1}T(n-(k-1)) + (2^{k-1} - 1) & [\text{inductive hypothesis}] \\ &= 2^{k-1}(2T(n-k) + 1) + (2^{k-1} - 1) & [\text{initial recurrence for } T(n-(k-1))] \\ &= 2^kT(n-k) + (2^k - 1) \quad \checkmark & [\text{algebra}] \end{aligned}$$

Our guess was correct! In particular, unrolling the recurrence n times give us the recurrence $T(n) = 2^n T(0) + (2^n - 1)$. Plugging in the base case $T(0) = 0$ give us the closed-form solution $T(n) = 2^n - 1$.

2.2 Fibonacci numbers

Let's try a less trivial example: the Fibonacci numbers $F_n = F_{n-1} + F_{n-2}$ with base cases $F_0 = 0$ and $F_1 = 1$. There is no obvious pattern in the first several values (aside from the recurrence itself), but we can reasonably guess that F_n is exponential in n . Let's try to prove inductively that $F_n \leq \alpha \cdot c^n$ for some constants $\alpha > 0$ and $c > 1$ and see how far we get.

$$\begin{aligned} F_n &= F_{n-1} + F_{n-2} \\ &\leq \alpha \cdot c^{n-1} + \alpha \cdot c^{n-2} & [\text{"induction hypothesis"}] \\ &\leq \alpha \cdot c^n \quad ??? \end{aligned}$$

The last inequality is satisfied if $c^n \geq c^{n-1} + c^{n-2}$, or more simply, if $c^2 - c - 1 \geq 0$. The smallest value of c that works is $\phi = (1 + \sqrt{5})/2 \approx 1.618034$; the other root of the quadratic equation has smaller absolute value, so we can ignore it.

So we have *most* of an inductive proof that $F_n \leq \alpha \cdot \phi^n$ for *some* constant α . All that we're missing are the base cases, which (we can easily guess) must determine the value of the coefficient α . We quickly compute

$$\frac{F_0}{\phi^0} = \frac{0}{1} = 0 \quad \text{and} \quad \frac{F_1}{\phi^1} = \frac{1}{\phi} \approx 0.618034 > 0,$$

so the base cases of our induction proof are correct as long as $\alpha \geq 1/\phi$. It follows that $F_n \leq \phi^{n-1}$ for all $n \geq 0$.

What about a matching lower bound? Essentially the same inductive proof implies that $F_n \geq \beta \cdot \phi^n$ for some constant β , but the only value of β that works for *all* n is the trivial $\beta = 0$! We could try to find some lower-order term that makes the base case non-trivial, but an easier approach is to recall that

asymptotic $\Omega()$ bounds only have to work for *sufficiently large* n . So let's ignore the trivial base case $F_0 = 0$ and assume that $F_2 = 1$ is a base case instead. Some more easy calculation gives us

$$\frac{F_2}{\phi^2} = \frac{1}{\phi^2} \approx 0.381966 < \frac{1}{\phi}.$$

Thus, the new base cases of our induction proof are correct as long as $\beta \leq 1/\phi^2$, which implies that $F_n \geq \phi^{n-2}$ for all $n \geq 1$.

Putting the upper and lower bounds together, we obtain the tight asymptotic bound $F_n = \Theta(\phi^n)$. It is possible to get a more exact solution by speculatively refining and conforming our current bounds, but it's not easy. Fortunately, if we really need it, we can get an exact solution using the *annihilator* method, which we'll see later in these notes.

2.3 Mergesort

Mergesort is a classical recursive divide-and-conquer algorithm for sorting an array. The algorithm splits the array in half, recursively sorts the two halves, and then merges the two sorted subarrays into the final sorted array.

```
MERGESORT(A[1..n]):
  if (n > 1)
    m ← ⌊n/2⌋
    MERGESORT(A[1..m])
    MERGESORT(A[m+1..n])
    MERGE(A[1..n], m)
```

```
MERGE(A[1..n], m):
  i ← 1; j ← m + 1
  for k ← 1 to n
    if j > n
      B[k] ← A[i]; i ← i + 1
    else if i > m
      B[k] ← A[j]; j ← j + 1
    else if A[i] < A[j]
      B[k] ← A[i]; i ← i + 1
    else
      B[k] ← A[j]; j ← j + 1
  for k ← 1 to n
    A[k] ← B[k]
```

Let $T(n)$ denote the worst-case running time of MERGESORT when the input array has size n . The MERGE subroutine clearly runs in $\Theta(n)$ time, so the function $T(n)$ satisfies the following recurrence:

$$T(n) = \begin{cases} \Theta(1) & \text{if } n = 1, \\ T(\lceil n/2 \rceil) + T(\lfloor n/2 \rfloor) + \Theta(n) & \text{otherwise.} \end{cases}$$

For now, let's consider the special case where n is a power of 2; this assumption allows us to take the floors and ceilings out of the recurrence. (We'll see how to deal with the floors and ceilings later; the short version is that they don't matter.)

Because the recurrence itself is given only asymptotically—in terms of $\Theta()$ expressions—we can't hope for anything but an asymptotic solution. So we can safely simplify the recurrence further by removing the Θ 's; any asymptotic solution to the simplified recurrence will also satisfy the original recurrence. (This simplification is actually important for another reason; if we kept the asymptotic expressions, we might be tempted to simplify them inappropriately.)

Our simplified recurrence now looks like this:

$$T(n) = \begin{cases} 1 & \text{if } n = 1, \\ 2T(n/2) + n & \text{otherwise.} \end{cases}$$

To guess at a solution, let's try unrolling the recurrence.

$$\begin{aligned} T(n) &= 2T(n/2) + n \\ &= 2(2T(n/4) + n/2) + n \\ &= 4T(n/4) + 2n \\ &= 8T(n/8) + 3n = \dots \end{aligned}$$

It looks like $T(n)$ satisfies the recurrence $T(n) = 2^k T(n/2^k) + kn$ for any positive integer k . Let's verify this by induction.

$$\begin{aligned} T(n) &= 2T(n/2) + n = 2^1 T(n/2^1) + 1 \cdot n \quad \checkmark & [k = 1, \text{ given recurrence}] \\ T(n) &= 2^{k-1} T(n/2^{k-1}) + (k-1)n & [\text{inductive hypothesis}] \\ &= 2^{k-1} (2T(n/2^k) + n/2^{k-1}) + (k-1)n & [\text{substitution}] \\ &= 2^k T(n/2^k) + kn \quad \checkmark & [\text{algebra}] \end{aligned}$$

Our guess was right! The recurrence becomes trivial when $n/2^k = 1$, or equivalently, when $k = \log_2 n$:

$$T(n) = nT(1) + n \log_2 n = n \log_2 n + n.$$

Finally, we have to put back the Θ 's we stripped off; our final closed-form solution is $T(n) = \Theta(n \log n)$.

2.4 An uglier divide-and-conquer example

Consider the divide-and-conquer recurrence $T(n) = \sqrt{n} \cdot T(\sqrt{n}) + n$. This doesn't fit into the form required by the Master Theorem (which we'll see below), but it still sort of resembles the Mergesort recurrence—the total size of the subproblems at the first level of recursion is n —so let's guess that $T(n) = O(n \log n)$, and then try to prove that our guess is correct. (We could also attack this recurrence by unrolling, but let's see how far just guessing will take us.)

Let's start by trying to prove an upper bound $T(n) \leq a n \lg n$ for all sufficiently large n and some constant a to be determined later:

$$\begin{aligned} T(n) &= \sqrt{n} \cdot T(\sqrt{n}) + n \\ &\leq \sqrt{n} \cdot a \sqrt{n} \lg \sqrt{n} + n & [\text{induction hypothesis}] \\ &= (a/2)n \lg n + n & [\text{algebra}] \\ &\leq an \lg n \quad \checkmark & [\text{algebra}] \end{aligned}$$

The last inequality assumes only that $1 \leq (a/2) \log n$, or equivalently, that $n \geq 2^{2/a}$. In other words, the induction proof is correct if n is sufficiently large. So we were right!

But before you break out the champagne, what about the multiplicative constant a ? The proof worked for any constant a , no matter how small. This strongly suggests that our upper bound $T(n) = O(n \log n)$ is not tight. Indeed, if we try to prove a matching lower bound $T(n) \geq b n \log n$ for sufficiently large n , we run into trouble.

$$\begin{aligned} T(n) &= \sqrt{n} \cdot T(\sqrt{n}) + n \\ &\geq \sqrt{n} \cdot b \sqrt{n} \log \sqrt{n} + n & [\text{induction hypothesis}] \\ &= (b/2)n \log n + n \\ &\not\geq bn \log n \end{aligned}$$

The last inequality would be correct only if $1 > (b/2) \log n$, but that inequality is false for large values of n , no matter which constant b we choose.

Okay, so $\Theta(n \log n)$ is too big. How about $\Theta(n)$? The lower bound is easy to prove directly:

$$T(n) = \sqrt{n} \cdot T(\sqrt{n}) + n \geq n \checkmark$$

But an inductive proof of the upper bound fails.

$$\begin{aligned} T(n) &= \sqrt{n} \cdot T(\sqrt{n}) + n \\ &\leq \sqrt{n} \cdot a \sqrt{n} + n && [\text{induction hypothesis}] \\ &= (a+1)n && [\text{algebra}] \\ &\not\leq an \end{aligned}$$

Hmmm. So what's bigger than n and smaller than $n \lg n$? How about $n \sqrt{\lg n}$?

$$\begin{aligned} T(n) &= \sqrt{n} \cdot T(\sqrt{n}) + n \leq \sqrt{n} \cdot a \sqrt{n} \sqrt{\lg \sqrt{n}} + n && [\text{induction hypothesis}] \\ &= (a/\sqrt{2}) n \sqrt{\lg n} + n && [\text{algebra}] \\ &\leq a n \sqrt{\lg n} \quad \text{for large enough } n \checkmark \end{aligned}$$

Okay, the upper bound checks out; how about the lower bound?

$$\begin{aligned} T(n) &= \sqrt{n} \cdot T(\sqrt{n}) + n \geq \sqrt{n} \cdot b \sqrt{n} \sqrt{\lg \sqrt{n}} + n && [\text{induction hypothesis}] \\ &= (b/\sqrt{2}) n \sqrt{\lg n} + n && [\text{algebra}] \\ &\not\geq b n \sqrt{\lg n} \end{aligned}$$

No, the last step doesn't work. So $\Theta(n \sqrt{\lg n})$ doesn't work.

Okay... what else is between n and $n \lg n$? How about $n \lg \lg n$?

$$\begin{aligned} T(n) &= \sqrt{n} \cdot T(\sqrt{n}) + n \leq \sqrt{n} \cdot a \sqrt{n} \lg \lg \sqrt{n} + n && [\text{induction hypothesis}] \\ &= a n \lg \lg n - a n + n && [\text{algebra}] \\ &\leq a n \lg \lg n \quad \text{if } a \geq 1 \checkmark \end{aligned}$$

Hey look at that! For once, our upper bound proof requires a constraint on the hidden constant a . This is a good indication that we've found the right answer. Let's try the lower bound:

$$\begin{aligned} T(n) &= \sqrt{n} \cdot T(\sqrt{n}) + n \geq \sqrt{n} \cdot b \sqrt{n} \lg \lg \sqrt{n} + n && [\text{induction hypothesis}] \\ &= b n \lg \lg n - b n + n && [\text{algebra}] \\ &\geq b n \lg \lg n \quad \text{if } b \leq 1 \checkmark \end{aligned}$$

Hey, it worked! We have most of an inductive proof that $T(n) \leq an \lg \lg n$ for any $a \geq 1$ and most of an inductive proof that $T(n) \geq bn \lg \lg n$ for any $b \leq 1$. Technically, we're still missing the base cases in both proofs, but we can be fairly confident at this point that $T(n) = \Theta(n \log \log n)$.

3 Divide and Conquer Recurrences (Recursion Trees)

Many divide and conquer algorithms give us running-time recurrences of the form

$$T(n) = a T(n/b) + f(n) \quad (1)$$

where a and b are constants and $f(n)$ is some other function. There is a simple and general technique for solving many recurrences in this and similar forms, using a **recursion tree**. The root of the recursion tree is a box containing the value $f(n)$; the root has a children, each of which is the root of a (recursively defined) recursion tree for the function $T(n/b)$.

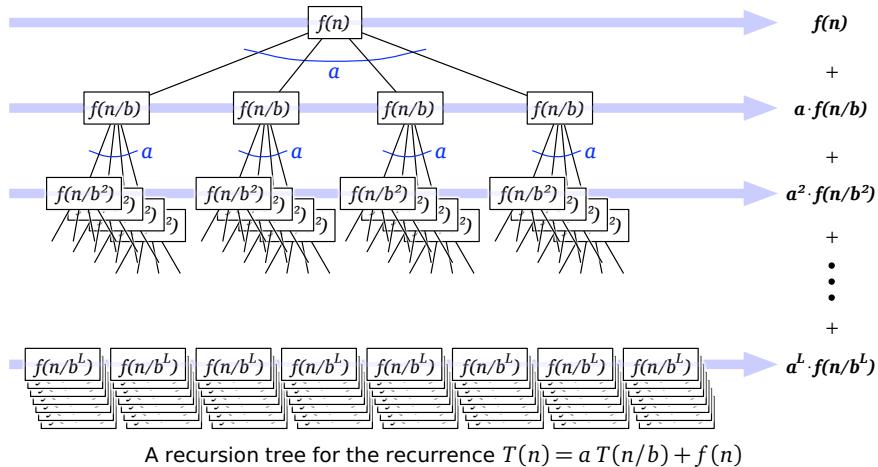
Equivalently, a recursion tree is a complete a -ary tree where each node at depth i contains the value $f(n/b^i)$. The recursion stops when we get to the base case(s) of the recurrence. Because we're only looking for asymptotic bounds, the exact base case doesn't matter; we can safely assume that $T(1) = \Theta(1)$, or even that $T(n) = \Theta(1)$ for all $n \leq 10^{100}$. I'll also assume for simplicity that n is an integral power of b ; we'll see how to avoid this assumption later (but to summarize: it doesn't matter).

Now $T(n)$ is just the sum of all values stored in the recursion tree. For each i , the i th level of the tree contains a^i nodes, each with value $f(n/b^i)$. Thus,

$$T(n) = \sum_{i=0}^L a^i f(n/b^i) \quad (\Sigma)$$

where L is the depth of the recursion tree. We easily see that $L = \log_b n$, because $n/b^L = 1$. The base case $f(1) = \Theta(1)$ implies that the last non-zero term in the summation is $\Theta(a^L) = \Theta(a^{\log_b n}) = \Theta(n^{\log_b a})$.

For most divide-and-conquer recurrences, the level-by-level sum (??) is a geometric series—each term is a constant factor larger or smaller than the previous term. In this case, only the largest term in the geometric series matters; all of the other terms are swallowed up by the $\Theta(\cdot)$ notation.



Here are several examples of the recursion-tree technique in action:

- **Mergesort (simplified):** $T(n) = 2T(n/2) + n$

There are 2^i nodes at level i , each with value $n/2^i$, so every term in the level-by-level sum (??) is the same:

$$T(n) = \sum_{i=0}^L n.$$

The recursion tree has $L = \log_2 n$ levels, so $T(n) = \Theta(n \log n)$.

- **Randomized selection:** $T(n) = T(3n/4) + n$

The recursion tree is a single path. The node at depth i has value $(3/4)^i n$, so the level-by-level sum (??) is a decreasing geometric series:

$$T(n) = \sum_{i=0}^L (3/4)^i n.$$

This geometric series is dominated by its initial term n , so $T(n) = \Theta(n)$. The recursion tree has $L = \log_{4/3} n$ levels, but so what?

- **Karatsuba's multiplication algorithm:** $T(n) = 3T(n/2) + n$

There are 3^i nodes at depth i , each with value $n/2^i$, so the level-by-level sum (??) is an increasing geometric series:

$$T(n) = \sum_{i=0}^L (3/2)^i n.$$

This geometric series is dominated by its final term $(3/2)^L n$. Each leaf contributes 1 to this term; thus, the final term is equal to the number of leaves in the tree! The recursion tree has $L = \log_2 n$ levels, and therefore $3^{\log_2 n} = n^{\log_2 3}$ leaves, so $T(n) = \Theta(n^{\log_2 3})$.

- $T(n) = 2T(n/2) + n / \lg n$

The sum of all the nodes in the i th level is $n/(\lg n - i)$. This implies that the depth of the tree is at most $\lg n - 1$. The level sums are neither constant nor a geometric series, so we just have to evaluate the overall sum directly.

Recall (or if you're seeing this for the first time: Behold!) that the n th *harmonic number* H_n is the sum of the reciprocals of the first n positive integers:

$$H_n := \sum_{i=1}^n \frac{1}{i}$$

It's not hard to show that $H_n = \Theta(\log n)$; in fact, we have the stronger inequalities $\ln(n+1) \leq H_n \leq \ln n + 1$.

$$T(n) = \sum_{i=0}^{\lg n - 1} \frac{n}{\lg n - i} = \sum_{j=1}^{\lg n} \frac{n}{j} = nH_{\lg n} = \Theta(n \lg \lg n)$$

- $T(n) = 4T(n/2) + n \lg n$

There are 4^i nodes at each level i , each with value $(n/2^i) \lg(n/2^i) = (n/2^i)(\lg n - i)$; again, the depth of the tree is at most $\lg n - 1$. We have the following summation:

$$T(n) = \sum_{i=0}^{\lg n - 1} n2^i (\lg n - i)$$

We can simplify this sum by substituting $j = \lg n - i$:

$$T(n) = \sum_{j=i}^{\lg n} n2^{\lg n - j} j = \sum_{j=i}^{\lg n} n^2 j / 2^j = \Theta(j^2)$$

The last step uses the fact that $\sum_{i=1}^{\infty} j/2^j = 2$. Although this is not quite a geometric series, it is still dominated by its largest term.

- **Ugly divide and conquer:** $T(n) = \sqrt{n} \cdot T(\sqrt{n}) + n$

We solved this recurrence earlier by guessing the right answer and verifying, but we can use recursion trees to get the correct answer directly. The *degree* of the nodes in the recursion tree is no longer constant, so we have to be a bit more careful, but the same basic technique still applies. It's not hard to see that the nodes in any level sum to n . The depth L satisfies the identity $n^{2^{-L}} = 2$ (we can't get all the way down to 1 by taking square roots), so $L = \lg \lg n$ and $T(n) = \Theta(n \lg \lg n)$.

- **Randomized quicksort:** $T(n) = T(3n/4) + T(n/4) + n$

This recurrence isn't in the standard form described earlier, but we can still solve it using recursion trees. Now nodes in the same level of the recursion tree have different values, and different leaves are at different levels. However, the nodes in any *complete* level (that is, above any of the leaves) sum to n . Moreover, every leaf in the recursion tree has depth between $\log_4 n$ and $\log_{4/3} n$. To derive an upper bound, we overestimate $T(n)$ by ignoring the base cases and extending the tree downward to the level of the *deepest* leaf. Similarly, to derive a lower bound, we overestimate $T(n)$ by counting only nodes in the tree up to the level of the *shallowest* leaf. These observations give us the upper and lower bounds $n \log_4 n \leq T(n) \leq n \log_{4/3} n$. Since these bounds differ by only a constant factor, we have $T(n) = \Theta(n \log n)$.

- **Deterministic selection:** $T(n) = T(n/5) + T(7n/10) + n$

Again, we have a lopsided recursion tree. If we look only at complete levels of the tree, we find that the level sums form a descending geometric series $T(n) = n + 9n/10 + 81n/100 + \dots$. We can get an upper bound by ignoring the base cases entirely and growing the tree out to infinity, and we can get a lower bound by only counting nodes in complete levels. Either way, the geometric series is dominated by its largest term, so $T(n) = \Theta(n)$.

- **Randomized search trees:** $T(n) = \frac{1}{4}T(n/4) + \frac{3}{4}T(3n/4) + 1$

This looks like a divide-and-conquer recurrence, but what does it mean to have a quarter of a child? The right approach is to imagine that each node in the recursion tree has a *weight* in addition to its value. Alternately, we get a standard recursion tree again if we add a second real parameter to the recurrence, defining $T(n) = T(n, 1)$, where

$$T(n, \alpha) = T(n/4, \alpha/4) + T(3n/4, 3\alpha/4) + \alpha.$$

In each complete level of the tree, the (weighted) node values sum to exactly 1. The leaves of the recursion tree are at different levels, but all between $\log_4 n$ and $\log_{4/3} n$. So we have upper and lower bounds $\log_4 n \leq T(n) \leq \log_{4/3} n$, which differ by only a constant factor, so $T(n) = \Theta(\log n)$.

- **Ham-sandwich trees:** $T(n) = T(n/2) + T(n/4) + 1$

Again, we have a lopsided recursion tree. If we only look at complete levels, we find that the level sums form an *ascending* geometric series $T(n) = 1 + 2 + 4 + \dots$, so the solution is dominated by the number of leaves. The recursion tree has $\log_4 n$ complete levels, so there are more than

$2^{\log_4 n} = n^{\log_4 2} = \sqrt{n}$; on the other hand, every leaf has depth at most $\log_2 n$, so the total number of leaves is at most $2^{\log_2 n} = n$. Unfortunately, the crude bounds $\sqrt{n} \ll T(n) \ll n$ are the best we can derive using the techniques we know so far!

The following theorem completely describes the solution for any divide-and-conquer recurrence in the ‘standard form’ $T(n) = aT(n/b) + f(n)$, where a and b are constants and $f(n)$ is a polynomial. This theorem allows us to bypass recursion trees for ‘standard’ recurrences, but many people (including Jeff) find it harder to remember than the more general recursion-tree technique. Your mileage may vary.

The Master Theorem. *The recurrence $T(n) = aT(n/b) + f(n)$ can be solved as follows.*

- If $a f(n/b) = \kappa f(n)$ for some constant $\kappa < 1$, then $T(n) = \Theta(f(n))$.
- If $a f(n/b) = K f(n)$ for some constant $K > 1$, then $T(n) = \Theta(n^{\log_b a})$.
- If $a f(n/b) = f(n)$, then $T(n) = \Theta(f(n) \log_b n)$.
- If none of these three cases apply, you’re on your own.

Proof: If $f(n)$ is a *constant factor larger* than $a f(b/n)$, then by induction, the sum is a descending geometric series. The sum of any geometric series is a constant times its largest term. In this case, the largest term is the first term $f(n)$.

If $f(n)$ is a *constant factor smaller* than $a f(b/n)$, then by induction, the sum is an ascending geometric series. The sum of any geometric series is a constant times its largest term. In this case, this is the last term, which by our earlier argument is $\Theta(n^{\log_b a})$.

Finally, if $a f(b/n) = f(n)$, then by induction, each of the $L+1$ terms in the sum is equal to $f(n)$. \square

*3.1 The Nuclear Bomb

Finally, let me describe *without proof* a powerful generalization of the recursion tree method, first published by Lebanese researchers Mohamad Akra and Louay Bazzi in 1998. Consider a general divide-and-conquer recurrence of the form

$$T(n) = \sum_{i=1}^k a_i T(n/b_i) + f(n),$$

where k is a constant, $a_i > 0$ and $b_i > 1$ are constants for all i , and $f(n) = \Omega(n^c)$ and $f(n) = O(n^d)$ for some constants $0 < c \leq d$. (As usual, we assume the standard base case $T(\Theta(1)) = \Theta(1)$.) Akra and Bazzi prove that this recurrence has the closed-form asymptotic solution

$$T(n) = \Theta\left(n^\rho \left(1 + \int_1^n \frac{f(u)}{u^{\rho+1}} du\right)\right),$$

where ρ is the unique real solution to the equation

$$\sum_{i=1}^k a_i / b_i^\rho = 1.$$

In particular, the Akra-Bazzi theorem immediately implies the following form of the Master Theorem:

$$T(n) = aT(n/b) + n^c \implies T(n) = \begin{cases} \Theta(n^{\log_b a}) & \text{if } c < \log_b a - \varepsilon \\ \Theta(n^c \log n) & \text{if } c = \log_b a \\ \Theta(n^c) & \text{if } c > \log_b a + \varepsilon \end{cases}$$

The Akra-Bazzi theorem does not require that the parameters a_i and b_i are integers, or even rationals; on the other hand, even when all parameters are integers, the characteristic equation $\sum_i a_i/b_i^\rho = 1$ may have no analytical solution.

Here are a few examples of recurrences that are difficult (or impossible) for recursion trees, but have easy solutions using the Akra-Bazzi theorem.

- **Randomized quicksort:** $T(n) = T(3n/4) + T(n/4) + n$

The equation $(3/4)^\rho + (1/4)^\rho = 1$ has the unique solution $\rho = 1$, and therefore

$$T(n) = \Theta\left(n\left(1 + \int_1^n \frac{1}{u} du\right)\right) = O(n \log n).$$

- **Deterministic selection:** $T(n) = T(n/5) + T(7n/10) + n$

The equation $(1/5)^\rho + (7/10)^\rho = 1$ has no analytical solution. However, we easily observe that $(1/5)^x + (7/10)^x$ is a decreasing function of x , and therefore $0 < \rho < 1$. Thus, we have

$$\int_1^n \frac{f(u)}{u^{\rho+1}} du = \int_1^n u^{-\rho} du = \frac{u^{1-\rho}}{1-\rho} \Big|_{u=1}^n = \frac{n^{1-\rho} - 1}{1-\rho} = \Theta(n^{1-\rho}),$$

and therefore

$$T(n) = \Theta(n^\rho \cdot (1 + \Theta(n^{1-\rho}))) = \Theta(n).$$

- **Randomized search trees:** $T(n) = \frac{1}{4}T(n/4) + \frac{3}{4}T(3n/4) + 1$

The equation $\frac{1}{4}(\frac{1}{4})^\rho + \frac{3}{4}(\frac{3}{4})^\rho = 1$ has the unique solution $\rho = 0$, and therefore

$$T(n) = \Theta\left(1 + \int_1^n \frac{1}{u} du\right) = \Theta(\log n).$$

- **Ham-sandwich trees:** $T(n) = T(n/2) + T(n/4) + 1$. Recall that we could only prove the very weak bounds $\sqrt{n} \ll T(n) \ll n$ using recursion trees. The equation $(1/2)^\rho + (1/4)^\rho = 1$ has the unique solution $\rho = \log_2((1 + \sqrt{5})/2) \approx 0.69424$, which can be obtained by setting $x = 2^\rho$ and solving for x . Thus, we have

$$\int_1^n \frac{1}{u^{\rho+1}} du = \frac{u^{-\rho}}{-\rho} \Big|_{u=1}^n = \frac{1 - n^{-\rho}}{\rho} = \Theta(1)$$

and therefore

$$T(n) = \Theta(n^\rho (1 + \Theta(1))) = \Theta(n^{\lg \phi}).$$

The obvious advantage of the Akra-Bazzi method is that it can solve *almost* any divide-and-conquer recurrence with just a few lines of calculation. (There are a few nasty exceptions like $T(n) = \sqrt{n} T(\sqrt{n}) + n$ where we have to fall back on recursion trees.) On the other hand, the steps appear to be magic, which makes the method hard to remember, and for *most* divide-and-conquer recurrences, the much simpler recursion tree method is sufficient.

4 Linear Recurrences (Annihilators)

Another common class of recurrences, called *linear* recurrences, arises in the context of recursive backtracking algorithms and counting problems. These recurrences express each function value $f(n)$ as a *linear* combination of a small number of nearby values $f(n-1), f(n-2), f(n-3), \dots$. The Fibonacci recurrence is a typical example:

$$F(n) = \begin{cases} 0 & \text{if } n = 0 \\ 1 & \text{if } n = 1 \\ F(n-1) + F(n-2) & \text{otherwise} \end{cases}$$

It turns out that the solution to *any* linear recurrence is a simple combination of polynomial and exponential functions in n . For example, we can verify by induction that the linear recurrence

$$T(n) = \begin{cases} 1 & \text{if } n = 0 \\ 0 & \text{if } n = 1 \text{ or } n = 2 \\ 3T(n-1) - 8T(n-2) + 4T(n-3) & \text{otherwise} \end{cases}$$

has the closed-form solution $T(n) = (n-3)2^n + 4$. First we check the base cases:

$$\begin{aligned} T(0) &= (0-3)2^0 + 4 = 1 & \checkmark \\ T(1) &= (1-3)2^1 + 4 = 0 & \checkmark \\ T(2) &= (2-3)2^2 + 4 = 0 & \checkmark \end{aligned}$$

And now the recursive case:

$$\begin{aligned} T(n) &= 3T(n-1) - 8T(n-2) + 4T(n-3) \\ &= 3((n-4)2^{n-1} + 4) - 8((n-5)2^{n-2} + 4) + 4((n-6)2^{n-3} + 4) \\ &= \left(\frac{3}{2} - \frac{8}{4} + \frac{4}{8}\right)n \cdot 2^n - \left(\frac{12}{2} - \frac{40}{4} + \frac{24}{8}\right)2^n + (2-8+4) \cdot 4 \\ &= (n-3) \cdot 2^n + 4 & \checkmark \end{aligned}$$

But how could we have possibly come up with that solution? In this section, I'll describe a general method for solving linear recurrences that's arguably easier than the induction proof!

4.1 Operators

Our technique for solving linear recurrences relies on the theory of *operators*. Operators are higher-order functions, which take one or more functions as input and produce different functions as output. For example, your first two semesters of calculus focus almost exclusively on the *differential* and *integral* operators $\frac{d}{dx}$ and $\int dx$. All the operators we will need are combinations of three elementary building blocks:

- **Sum:** $(f + g)(n) := f(n) + g(n)$
- **Scale:** $(\alpha \cdot f)(n) := \alpha \cdot (f(n))$
- **Shift:** $(Ef)(n) := f(n+1)$

The shift and scale operators are *linear*, which means they can be distributed over sums; for example, for any functions f , g , and h , we have $E(f - 3(g - h)) = Ef + (-3)Eg + 3 Eh$.

We can combine these building blocks to obtain more complex *compound* operators. For example, the compound operator $E - 2$ is defined by setting $(E - 2)f := Ef + (-2)f$ for any function f . We can also apply the shift operator twice: $(E(Ef))(n) = f(n+2)$; we write usually E^2f as a synonym for $E(Ef)$. More generally, for any positive integer k , the operator E^k shifts its argument k times: $E^k f(n) = f(n+k)$. Similarly, $(E - 2)^2$ is shorthand for the operator $(E - 2)(E - 2)$, which applies $(E - 2)$ twice.

For example, here are the results of applying different operators to the function $f(n) = 2^n$:

$$\begin{aligned} 2f(n) &= 2 \cdot 2^n = 2^{n+1} \\ 3f(n) &= 3 \cdot 2^n \\ Ef(n) &= 2^{n+1} \\ E^2f(n) &= 2^{n+2} \\ (E - 2)f(n) &= Ef(n) - 2f(n) = 2^{n+1} - 2^{n+1} = 0 \\ (E^2 - 1)f(n) &= E^2f(n) - f(n) = 2^{n+2} - 2^n = 3 \cdot 2^n \end{aligned}$$

These compound operators can be manipulated exactly as though they were polynomials over the ‘variable’ E . In particular, we can ‘factor’ compound operators into ‘products’ of simpler operators, and the order of the factors is unimportant. For example, the compound operators $E^2 - 3E + 2$ and $(E - 1)(E - 2)$ are equivalent:

$$\text{Let } g(n) := (E - 2)f(n) = f(n+1) - 2f(n).$$

$$\begin{aligned} \text{Then } (E - 1)(E - 2)f(n) &= (E - 1)g(n) \\ &= g(n+1) - g(n) \\ &= (f(n+2) - 2f(n-1)) - (f(n+1) - 2f(n)) \\ &= f(n+2) - 3f(n+1) + 2f(n) \\ &= (E^2 - 3E + 2)f(n). \quad \checkmark \end{aligned}$$

It is an easy exercise to confirm that $E^2 - 3E + 2$ is also equivalent to the operator $(E - 2)(E - 1)$.

The following table summarizes everything we need to remember about operators.

Operator	Definition
addition	$(f + g)(n) := f(n) + g(n)$
subtraction	$(f - g)(n) := f(n) - g(n)$
multiplication	$(\alpha \cdot f)(n) := \alpha \cdot (f(n))$
shift	$Ef(n) := f(n+1)$
k -fold shift	$E^k f(n) := f(n+k)$
composition	$(X + Y)f := Xf + Yf$ $(X - Y)f := Xf - Yf$ $XYf := X(Yf) = Y(Xf)$
distribution	$X(f + g) = Xf + Xg$

4.2 Annihilators

An *annihilator* of a function f is any nontrivial operator that transforms f into the zero function. (We can trivially annihilate any function by multiplying it by zero, so as a technical matter, we do not consider

the zero operator to be an annihilator.) Every compound operator we consider annihilates a specific class of functions; conversely, every function composed of polynomial and exponential functions has a unique (minimal) annihilator.

We have already seen that the operator $(E - 2)$ annihilates the function 2^n . It's not hard to see that the operator $(E - c)$ annihilates the function $\alpha \cdot c^n$, for any constants c and α . More generally, the operator $(E - c)$ annihilates the function a^n if and only if $c = a$:

$$(E - c)a^n = Ea^n - c \cdot a^n = a^{n+1} - c \cdot a^n = (a - c)a^n.$$

Thus, $(E - 2)$ is essentially the *only* annihilator of the function 2^n .

What about the function $2^n + 3^n$? The operator $(E - 2)$ annihilates the function 2^n , but leaves the function 3^n unchanged. Similarly, $(E - 3)$ annihilates 3^n while *negating* the function 2^n . But if we apply *both* operators, we annihilate both terms:

$$\begin{aligned} (E - 2)(2^n + 3^n) &= E(2^n + 3^n) - 2(2^n + 3^n) \\ &= (2^{n+1} + 3^{n+1}) - (2^{n+1} + 2 \cdot 3^n) = 3^n \\ \implies (E - 3)(E - 2)(2^n + 3^n) &= (E - 3)3^n = 0 \end{aligned}$$

In general, for any integers $a \neq b$, the operator $(E - a)(E - b) = (E - b)(E - a) = (E^2 - (a + b)E + ab)$ annihilates any function of the form $\alpha a^n + \beta b^n$, but nothing else.

What about the operator $(E - a)(E - a) = (E - a)^2$? It turns out that this operator annihilates all functions of the form $(\alpha n + \beta)a^n$:

$$\begin{aligned} (E - a)((\alpha n + \beta)a^n) &= (\alpha(n+1) + \beta)a^{n+1} - a(\alpha n + \beta)a^n \\ &= \alpha a^{n+1} \\ \implies (E - a)^2((\alpha n + \beta)a^n) &= (E - a)(\alpha a^{n+1}) = 0 \end{aligned}$$

More generally, the operator $(E - a)^d$ annihilates all functions of the form $p(n) \cdot a^n$, where $p(n)$ is a polynomial of degree at most $d - 1$. For example, $(E - 1)^3$ annihilates any polynomial of degree at most 2.

The following table summarizes everything we need to remember about annihilators.

Operator	Functions annihilated
$E - 1$	α
$E - a$	αa^n
$(E - a)(E - b)$	$\alpha a^n + \beta b^n$ [if $a \neq b$]
$(E - a_0)(E - a_1) \cdots (E - a_k)$	$\sum_{i=0}^k \alpha_i a_i^n$ [if a_i distinct]
$(E - 1)^2$	$\alpha n + \beta$
$(E - a)^2$	$(\alpha n + \beta)a^n$
$(E - a)^2(E - b)$	$(\alpha n + \beta)a^b + \gamma b^n$ [if $a \neq b$]
$(E - a)^d$	$(\sum_{i=0}^{d-1} \alpha_i n^i)a^n$
If X annihilates f , then X also annihilates Ef .	
If X annihilates both f and g , then X also annihilates $f \pm g$.	
If X annihilates f , then X also annihilates αf , for any constant α .	
If X annihilates f and Y annihilates g , then XY annihilates $f \pm g$.	

4.3 Annihilating Recurrences

Given a linear recurrence for a function, it's easy to extract an annihilator for that function. For many recurrences, we only need to rewrite the recurrence in operator notation. Once we have an annihilator, we can factor it into operators of the form $(E - c)$; the table on the previous page then gives us a generic solution with some unknown coefficients. If we are given explicit base cases, we can determine the coefficients by examining a few small cases; in general, this involves solving a small system of linear equations. If the base cases are not specified, the generic solution almost always gives us an asymptotic solution. Here is the technique step by step:

1. Write the recurrence in operator form
2. Extract an annihilator for the recurrence
3. Factor the annihilator (if necessary)
4. Extract the *generic solution* from the annihilator
5. Solve for coefficients using base cases (if known)

Here are several examples of the technique in action:

- $r(n) = 5r(n - 1)$, where $r(0) = 3$.

1. We can write the recurrence in operator form as follows:

$$r(n) = 5r(n - 1) \implies r(n + 1) - 5r(n) = 0 \implies (E - 5)r(n) = 0.$$

2. We immediately see that $(E - 5)$ annihilates the function $r(n)$.
3. The annihilator $(E - 5)$ is already factored.
4. Consulting the annihilator table on the previous page, we find the generic solution $r(n) = \alpha 5^n$ for some constant α .
5. The base case $r(0) = 3$ implies that $\alpha = 3$.

We conclude that $r(n) = 3 \cdot 5^n$. We can easily verify this closed-form solution by induction:

$$\begin{aligned} r(0) &= 3 \cdot 5^0 = 3 \quad \checkmark && [\text{definition}] \\ r(n) &= 5r(n - 1) && [\text{definition}] \\ &= 5 \cdot (3 \cdot 5^{n-1}) && [\text{induction hypothesis}] \\ &= 5^n \cdot 3 \quad \checkmark && [\text{algebra}] \end{aligned}$$

- **Fibonacci numbers:** $F(n) = F(n - 1) + F(n - 2)$, where $F(0) = 0$ and $F(1) = 1$.

1. We can rewrite the recurrence as $(E^2 - E - 1)F(n) = 0$.
2. The operator $E^2 - E - 1$ clearly annihilates $F(n)$.
3. The quadratic formula implies that the annihilator $E^2 - E - 1$ factors into $(E - \phi)(E - \hat{\phi})$, where $\phi = (1 + \sqrt{5})/2 \approx 1.618034$ is the golden ratio and $\hat{\phi} = (1 - \sqrt{5})/2 = 1 - \phi = -1/\phi$.
4. The annihilator implies that $F(n) = \alpha\phi^n + \hat{\alpha}\hat{\phi}^n$ for some unknown constants α and $\hat{\alpha}$.

5. The base cases give us two equations in two unknowns:

$$\begin{aligned} F(0) &= 0 = \alpha + \hat{\alpha} \\ F(1) &= 1 = \alpha\phi + \hat{\alpha}\hat{\phi} \end{aligned}$$

Solving this system of equations gives us $\alpha = 1/(2\phi - 1) = 1/\sqrt{5}$ and $\hat{\alpha} = -1/\sqrt{5}$.

We conclude with the following exact closed form for the n th Fibonacci number:

$$F(n) = \frac{\phi^n - \hat{\phi}^n}{\sqrt{5}} = \frac{1}{\sqrt{5}} \left(\frac{1 + \sqrt{5}}{2} \right)^n - \frac{1}{\sqrt{5}} \left(\frac{1 - \sqrt{5}}{2} \right)^n$$

With all the square roots in this formula, it's quite amazing that Fibonacci numbers are integers. However, if we do all the math correctly, all the square roots cancel out when i is an integer. (In fact, this is pretty easy to prove using the binomial theorem.)

- **Towers of Hanoi:** $T(n) = 2T(n - 1) + 1$, where $T(0) = 0$. This is our first example of a *non-homogeneous* recurrence, which means the recurrence has one or more non-recursive terms.

1. We can rewrite the recurrence as $(E - 2)T(n) = 1$.
2. The operator $(E - 2)$ doesn't quite annihilate the function; it leaves a *residue* of 1. But we can annihilate the residue by applying the operator $(E - 1)$. Thus, the compound operator $(E - 1)(E - 2)$ annihilates the function.
3. The annihilator is already factored.
4. The annihilator table gives us the generic solution $T(n) = \alpha 2^n + \beta$ for some unknown constants α and β .
5. The base cases give us $T(0) = 0 = \alpha 2^0 + \beta$ and $T(1) = 1 = \alpha 2^1 + \beta$. Solving this system of equations, we find that $\alpha = 1$ and $\beta = -1$.

We conclude that $T(n) = 2^n - 1$.

For the remaining examples, I won't explicitly enumerate the steps in the solution.

- **Height-balanced trees:** $H(n) = H(n - 1) + H(n - 2) + 1$, where $H(-1) = 0$ and $H(0) = 1$. (Yes, we're starting at -1 instead of 0 . So what?)

We can rewrite the recurrence as $(E^2 - E - 1)H = 1$. The residue 1 is annihilated by $(E - 1)$, so the compound operator $(E - 1)(E^2 - E - 1)$ annihilates the recurrence. This operator factors into $(E - 1)(E - \phi)(E - \hat{\phi})$, where $\phi = (1 + \sqrt{5})/2$ and $\hat{\phi} = (1 - \sqrt{5})/2$. Thus, we get the generic solution $H(n) = \alpha \cdot \phi^n + \beta + \gamma \cdot \hat{\phi}^n$, for some unknown constants α, β, γ that satisfy the following system of equations:

$$\begin{aligned} H(-1) &= 0 = \alpha\phi^{-1} + \beta + \gamma\hat{\phi}^{-1} = \alpha/\phi + \beta - \gamma/\hat{\phi} \\ H(0) &= 1 = \alpha\phi^0 + \beta + \gamma\hat{\phi}^0 = \alpha + \beta + \gamma \\ H(1) &= 2 = \alpha\phi^1 + \beta + \gamma\hat{\phi}^1 = \alpha\phi + \beta + \gamma\hat{\phi} \end{aligned}$$

Solving this system (using Cramer's rule or Gaussian elimination), we find that $\alpha = (\sqrt{5} + 2)/\sqrt{5}$, $\beta = -1$, and $\gamma = (\sqrt{5} - 2)/\sqrt{5}$. We conclude that

$$H(n) = \frac{\sqrt{5} + 2}{\sqrt{5}} \left(\frac{1 + \sqrt{5}}{2} \right)^n - 1 + \frac{\sqrt{5} - 2}{\sqrt{5}} \left(\frac{1 - \sqrt{5}}{2} \right)^n.$$

- $T(n) = 3T(n-1) - 8T(n-2) + 4T(n-3)$, where $T(0) = 1$, $T(1) = 0$, and $T(2) = 0$. This was our original example of a linear recurrence.

We can rewrite the recurrence as $(E^3 - 3E^2 + 8E - 4)T = 0$, so we immediately have an annihilator $E^3 - 3E^2 + 8E - 4$. Using high-school algebra, we can factor the annihilator into $(E - 2)^2(E - 1)$, which implies the generic solution $T(n) = \alpha n2^n + \beta 2^n + \gamma$. The constants α , β , and γ are determined by the base cases:

$$\begin{aligned} T(0) &= 1 = \alpha \cdot 0 \cdot 2^0 + \beta 2^0 + \gamma = \beta + \gamma \\ T(1) &= 0 = \alpha \cdot 1 \cdot 2^1 + \beta 2^1 + \gamma = 2\alpha + 2\beta + \gamma \\ T(2) &= 0 = \alpha \cdot 2 \cdot 2^2 + \beta 2^2 + \gamma = 8\alpha + 4\beta + \gamma \end{aligned}$$

Solving this system of equations, we find that $\alpha = 1$, $\beta = -3$, and $\gamma = 4$, so $T(n) = (n-3)2^n + 4$.

- $T(n) = T(n - 1) + 2T(n - 2) + 2^n - n^2$

We can rewrite the recurrence as $(E^2 - E - 2)T(n) = E^2(2^n - n^2)$. Notice that we had to shift up the non-recursive parts of the recurrence when we expressed it in this form. The operator $(E - 2)(E - 1)^3$ annihilates the residue $2^n - n^2$, and therefore also annihilates the shifted residue $E^2(2^n + n^2)$. Thus, the operator $(E - 2)(E - 1)^3(E^2 - E - 2)$ annihilates the entire recurrence. We can factor the quadratic factor into $(E - 2)(E + 1)$, so the annihilator factors into $(E - 2)^2(E - 1)^3(E + 1)$. So the generic solution is $T(n) = \alpha n 2^n + \beta 2^n + \gamma n^2 + \delta n + \epsilon + \eta(-1)^n$. The coefficients $\alpha, \beta, \gamma, \delta, \epsilon, \eta$ satisfy a system of six equations determined by the first six function values $T(0)$ through $T(5)$. For almost² every set of base cases, we have $\alpha \neq 0$, which implies that $T(n) = \Theta(n 2^n)$.

For a more detailed explanation of the annihilator method, see George Lueker, Some techniques for solving recurrences, *ACM Computing Surveys* 12(4):419-436, 1980.

5 Transformations

Sometimes we encounter recurrences that don't fit the structures required for recursion trees or annihilators. In many of those cases, we can transform the recurrence into a more familiar form, by defining a new function in terms of the one we want to solve. There are many different kinds of transformations, but these three are probably the most useful:

- Domain transformation: Define a new function $S(n) = T(f(n))$ with a simpler recurrence, for some simple function f .
- Range transformation: Define a new function $S(n) = f(T(n))$ with a simpler recurrence, for some simple function f .
- Difference transformation: Simplify the recurrence for $T(n)$ by considering the difference $T(n) - T(n - 1)$.

Here are some examples of these transformations in action.

- **Unsimplified Mergesort:** $T(n) = T(\lceil n/2 \rceil) + T(\lfloor n/2 \rfloor) + \Theta(n)$

When n is a power of 2, we can simplify the mergesort recurrence to $T(n) = 2T(n/2) + \Theta(n)$, which has the solution $T(n) = \Theta(n \log n)$. Unfortunately, for other values of n , this simplified recurrence is incorrect. When n is odd, then the recurrence calls for us to sort a fractional number of elements! Worse yet, if n is not a power of 2, we will *never* reach the base case $T(1) = 1$.

So we really need to solve the original recurrence. We have no hope of getting an *exact* solution, even if we ignore the $\Theta()$ in the recurrence; the floors and ceilings will eventually kill us. But we can derive a tight asymptotic solution using a domain transformation—we can rewrite the function $T(n)$ as a nested function $S(f(n))$, where $f(n)$ is a simple function and the function $S()$ has a simpler recurrence.

First let's overestimate the time bound, once by pretending that the two subproblem sizes are equal, and again to eliminate the ceiling:

$$T(n) \leq 2T(\lceil n/2 \rceil) + n \leq 2T(n/2 + 1) + n.$$

²In fact, the only possible solutions with $\alpha = 0$ have the form $-2^{n-1} - n^2/2 - 5n/2 + \eta(-1)^n$ for some constant η .

Now we define a new function $S(n) = T(n + \alpha)$, where α is an unknown constant, chosen so that $S(n)$ satisfies the Master-Theorem-ready recurrence $S(n) \leq 2S(n/2) + O(n)$. To figure out the correct value of α , we compare two versions of the recurrence for the function $T(n + \alpha)$:

$$\begin{aligned} S(n) &\leq 2S(n/2) + O(n) & \Rightarrow & \quad T(n + \alpha) \leq 2T(n/2 + \alpha) + O(n) \\ T(n) &\leq 2T(n/2 + 1) + n & \Rightarrow & \quad T(n + \alpha) \leq 2T((n + \alpha)/2 + 1) + n + \alpha \end{aligned}$$

For these two recurrences to be equal, we need $n/2 + \alpha = (n + \alpha)/2 + 1$, which implies that $\alpha = 2$. The Master Theorem now tells us that $S(n) = O(n \log n)$, so

$$T(n) = S(n - 2) = O((n - 2) \log(n - 2)) = O(n \log n).$$

A similar argument implies the matching lower bound $T(n) = \Omega(n \log n)$. So $T(n) = \Theta(n \log n)$ after all, just as though we had ignored the floors and ceilings from the beginning!

Domain transformations are useful for removing floors, ceilings, and lower order terms from the arguments of any recurrence that otherwise looks like it ought to fit either the Master Theorem or the recursion tree method. But now that we know this, we don't need to bother grinding through the actual gory details!

- **Ham-Sandwich Trees:** $T(n) = T(n/2) + T(n/4) + 1$

As we saw earlier, the recursion tree method only gives us the uselessly loose bounds $\sqrt{n} \ll T(n) \ll n$ for this recurrence, and the recurrence is in the wrong form for annihilators. The authors who discovered ham-sandwich trees (yes, this is a real data structure) solved this recurrence by guessing the solution and giving a complicated induction proof.

But a simple transformation allows us to solve the recurrence in just a few lines. We define a new function $t(k) = T(2^k)$, which satisfies the simpler linear recurrence $t(k) = t(k - 1) + t(k - 2) + 1$. This recurrence should immediately remind you of Fibonacci numbers. Sure enough, the annihilator method implies the solution $t(k) = \Theta(\phi^k)$, where $\phi = (1 + \sqrt{5})/2$ is the golden ratio. We conclude that

$$T(n) = t(\lg n) = \Theta(\phi^{\lg n}) = \Theta(n^{\lg \phi}) \approx \Theta(n^{0.69424}).$$

Recall that we obtained this same solution earlier using the Akra-Bazzi theorem.

Many other divide-and-conquer recurrences can be similarly transformed into linear recurrences and then solved with annihilators. Consider once more the simplified mergesort recurrence $T(n) = 2T(n/2) + n$. The function $t(k) = T(2^k)$ satisfies the recurrence $t(k) = 2t(k - 1) + 2^k$. The annihilator method gives us the generic solution $t(k) = \Theta(k \cdot 2^k)$, which implies that $T(n) = t(\lg n) = \Theta(n \log n)$, just as we expected.

On the other hand, for some recurrences like $T(n) = T(n/3) + T(2n/3) + n$, the recursion tree method gives an easy solution, but there's no way to transform the recurrence into a form where we can apply the annihilator method directly.³

³However, we can still get a solution via functional transformations as follows. The function $t(k) = T((3/2)^k)$ satisfies the recurrence $t(n) = t(n - 1) + t(n - \lambda) + (3/2)^k$, where $\lambda = \log_{3/2} 3 = 2.709511\dots$. The **characteristic function** for this recurrence is $(r^\lambda - r^{\lambda-1} - 1)(r - 3/2)$, which has a double root at $r = 3/2$ and nowhere else. Thus, $t(k) = \Theta(k(3/2)^k)$, which implies that $T(n) = t(\log_{3/2} n) = \Theta(n \log n)$. This line of reasoning is the core of the Akra-Bazzi method.

- **Random Binary Search Trees:** $T(n) = \frac{1}{4}T(n/4) + \frac{3}{4}T(3n/4) + 1$

This looks like a divide-and-conquer recurrence, so we might be tempted to apply recursion trees, but what does it mean to have a quarter of a child? If we're not comfortable with weighted recursion trees or the Akra-Bazzi theorem, we can instead consider a new function $U(n) = n \cdot T(n)$, which satisfies the more palatable recurrence $U(n) = U(n/4) + U(3n/4) + n$. As we've already seen, recursion trees imply that $U(n) = \Theta(n \log n)$, which immediately implies that $T(n) = \Theta(\log n)$.

- **Randomized Quicksort:** $T(n) = \frac{2}{n} \sum_{k=0}^{n-1} T(k) + n$

This is our first example of a *full history* recurrence; each function value $T(n)$ is defined in terms of *all* previous function values $T(k)$ with $k < n$. Before we can apply any of our existing techniques, we need to convert this recurrence into an equivalent *limited history* form by shifting and subtracting away common terms. To make this step slightly easier, we first multiply both sides of the recurrence by n to get rid of the fractions.

$$\begin{aligned} n \cdot T(n) &= 2 \sum_{k=0}^{n-1} T(k) + n^2 && [\text{multiply both sides by } n] \\ (n-1) \cdot T(n-1) &= 2 \sum_{k=0}^{n-2} T(k) + (n-1)^2 && [\text{shift}] \\ nT(n) - (n-1)T(n-1) &= 2T(n-1) + 2n - 1 && [\text{subtract}] \\ T(n) &= \frac{n+1}{n} T(n-1) + 2 - \frac{1}{n} && [\text{simplify}] \end{aligned}$$

We can solve this limited-history recurrence using another functional transformation. We define a new function $t(n) = T(n)/(n+1)$, which satisfies the simpler recurrence

$$t(n) = t(n-1) + \frac{2}{n+1} - \frac{1}{n(n+1)},$$

which we can easily unroll into a summation. If we only want an asymptotic solution, we can simplify the final recurrence to $t(n) = t(n-1) + \Theta(1/n)$, which unrolls into a very familiar summation:

$$t(n) = \sum_{i=1}^n \Theta(1/i) = \Theta(H_n) = \Theta(\log n).$$

Finally, substituting $T(n) = (n+1)t(n)$ gives us a solution to the original recurrence: $T(n) = \Theta(n \log n)$.

Exercises

1. For each of the following recurrences, first *guess* an exact closed-form solution, and then prove your guess is correct. You are free to use any method you want to make your guess—unrolling the recurrence, writing out the first several values, induction proof template, recursion trees, annihilators, transformations, ‘It looks like that other one’, whatever—but please describe your method. All functions are from the non-negative integers to the reals. If it simplifies your solutions, express them in terms of Fibonacci numbers F_n , harmonic numbers H_n , binomial coefficients $\binom{n}{k}$, factorials $n!$, and/or the floor and ceiling functions $\lfloor x \rfloor$ and $\lceil x \rceil$.

(a) $A(n) = A(n - 1) + 1$, where $A(0) = 0$.

(b) $B(n) = \begin{cases} 0 & \text{if } n < 5 \\ B(n - 5) + 2 & \text{otherwise} \end{cases}$

(c) $C(n) = C(n - 1) + 2n - 1$, where $C(0) = 0$.

(d) $D(n) = D(n - 1) + \binom{n}{2}$, where $D(0) = 0$.

(e) $E(n) = E(n - 1) + 2^n$, where $E(0) = 0$.

(f) $F(n) = 3 \cdot F(n - 1)$, where $F(0) = 1$.

(g) $G(n) = \frac{G(n-1)}{G(n-2)}$, where $G(0) = 1$ and $G(1) = 2$. [Hint: This is easier than it looks.]

(h) $H(n) = H(n - 1) + 1/n$, where $H(0) = 0$.

(i) $I(n) = I(n - 2) + 3/n$, where $I(0) = I(1) = 0$. [Hint: Consider even and odd n separately.]

(j) $J(n) = J(n - 1)^2$, where $J(0) = 2$.

(k) $K(n) = K(\lfloor n/2 \rfloor) + 1$, where $K(0) = 0$.

(l) $L(n) = L(n - 1) + L(n - 2)$, where $L(0) = 2$ and $L(1) = 1$.

[Hint: Write the solution in terms of Fibonacci numbers.]

(m) $M(n) = M(n - 1) \cdot M(n - 2)$, where $M(0) = 2$ and $M(1) = 1$.

[Hint: Write the solution in terms of Fibonacci numbers.]

(n) $N(n) = 1 + \sum_{k=1}^n (N(k - 1) + N(n - k))$, where $N(0) = 1$.

(p) $P(n) = \sum_{k=0}^{n-1} (k \cdot P(k - 1))$, where $P(0) = 1$.

(q) $Q(n) = \frac{1}{2 - Q(n-1)}$, where $Q(0) = 0$.

(r) $R(n) = \max_{1 \leq k \leq n} \{R(k - 1) + R(n - k) + n\}$

(s) $S(n) = \max_{1 \leq k \leq n} \{S(k - 1) + S(n - k) + 1\}$

(t) $T(n) = \min_{1 \leq k \leq n} \{T(k - 1) + T(n - k) + n\}$

(u) $U(n) = \min_{1 \leq k \leq n} \{U(k - 1) + U(n - k) + 1\}$

(v) $V(n) = \max_{n/3 \leq k \leq 2n/3} \{V(k - 1) + V(n - k) + n\}$

2. Use recursion trees or the Akra-Bazzi theorem to solve each of the following recurrences.

- (a) $A(n) = 2A(n/4) + \sqrt{n}$
- (b) $B(n) = 2B(n/4) + n$
- (c) $C(n) = 2C(n/4) + n^2$
- (d) $D(n) = 3D(n/3) + \sqrt{n}$
- (e) $E(n) = 3E(n/3) + n$
- (f) $F(n) = 3F(n/3) + n^2$
- (g) $G(n) = 4G(n/2) + \sqrt{n}$
- (h) $H(n) = 4H(n/2) + n$
- (i) $I(n) = 4I(n/2) + n^2$
- (j) $J(n) = J(n/2) + J(n/3) + J(n/6) + n$
- (k) $K(n) = K(n/2) + K(n/3) + K(n/6) + n^2$
- (l) $L(n) = L(n/15) + L(n/10) + 2L(n/6) + \sqrt{n}$
- *(m) $M(n) = 2M(n/3) + 2M(2n/3) + n$
- (n) $N(n) = \sqrt{2n}N(\sqrt{2n}) + \sqrt{n}$
- (p) $P(n) = \sqrt{2n}P(\sqrt{2n}) + n$
- (q) $Q(n) = \sqrt{2n}Q(\sqrt{2n}) + n^2$
- (r) $R(n) = R(n - 3) + 8^n$ — Don't use annihilators!
- (s) $S(n) = 2S(n - 2) + 4^n$ — Don't use annihilators!
- (t) $T(n) = 4T(n - 1) + 2^n$ — Don't use annihilators!

3. Make up a bunch of linear recurrences and then solve them using annihilators.

4. Solve the following recurrences, using any tricks at your disposal.

- (a) $T(n) = \sum_{i=1}^{\lg n} T(n/2^i) + n$ [Hint: Assume n is a power of 2.]
- (b) More to come...

Egzamin z matematyki dyskretnej

11 lutego 2020

- Niech $G = (V, E)$ oznacza graf, w którym $V = \{a_1, a_2, \dots, a_n, b_1, b_2, \dots, b_n\}$, wierzchołki a_1, a_2, \dots, a_n są połączone w cykl (tzn. dla każdego i , $1 \leq i < n$ wierzchołki a_i i a_{i+1} są połączone krawędzią oraz krawędziami są połączone a_1 i a_n), wierzchołki b_1, b_2, \dots, b_n również są połączone w cykl oraz każdy wierzchołek a_i jest połączony z każdym wierzchołkiem b_j .

Znajdź optymalne kolorowanie wierzchołkowe G . Potrzebne jest uzasadnienie.

- Wykaż, że liczba $53^{33} - 33^{33}$ jest podzielna przez 10.
- Znajdź ogólną postać rozwiązania następującego równania rekurencyjnego za pomocą anihilatorów:

$$a_{n+2} = 5a_{n+1} - 6a_n + \frac{\binom{n}{2}}{2^n}, \text{ gdy } a_0 = a_1 = 0.$$

- Podaj funkcję tworzącą dla ciągu $a_n = 1 + 2 + \dots + 2^n + (-\sqrt{2})^n$.
- W pewnej grupie muzykujących osób jedna gra na fortepianie, harfie i skrzypcach, druga na kontrabasie, harfie i wiolonczeli, trzecia na skrzypcach, czwarta na wiolonczeli i piąta na skrzypcach i wiolonczeli.

Chcieliby zagrać utwór na fortepian, skrzypce, wiolonczelę, kontrabas i harfę. Czy uda im się dobrze skłądzić?

- Niech G będzie grafem spójnym o m krawędziach. Dla każdej krawędzi e tego grafu mamy zadaną liczbę p_e oznaczającą wymaganą liczbę przejść tą krawędzią. Opracuj algorytm, który albo orzeka istnienie trasy w grafie G , w której każda krawędź jest strawersowana dokładnie p_e razy, albo stwierdza, że taka trasa nie istnieje. Punkt startu trasy nie musi być taki sam jak mety.

- Z macierzy $n \times n$ usuwamy część nad przekątną otrzymując macierz "schodkową". Na ile sposobów można ją podzielić na n prostokątów?
- Pewną grupę 24 osób składającą się z 12 kibiców drużyny A i 12 drużyn B chcemy rozmielić w czterech przedziałach 6-osobowych o numerach 1–4. Rozmieszczenia wewnętrz przedziałów są nieistotne.

- Na ile sposobów możemy to zrobić, aby w żadnym przedziale nie było tyle samo kibiców drużyny A co kibiców drużyny B ?
- Ile jest rozmieszczeń, w których sumaryczna liczba w przedziałach 1 i 2 kibiców A nie jest taka sama jak kibiców B ?

- Danych jest 12 różnych liczb dwucyfrowych. Wykaż, że wśród nich istnieją takie dwie, których różnica jest liczbą dwucyfrową o jednakowych cyfrach.

10. *Kwadratem łacińskim* nazywamy kwadrat $n \times n$, w którym na każdym polu stoi liczba ze zbioru $\{1, 2, \dots, n\}$ tak, że w każdej kolumnie oraz w każdym wierszu jest po jednej z liczb $\{1, 2, \dots, n\}$. *Prostokątem łacińskim* nazywamy prostokąt o n kolumnach i m wierszach, $1 \leq m \leq n$, w którym na każdym polu stoi liczba ze zbioru $\{1, 2, \dots, n\}$ tak, że w każdym wierszu każda z liczb $\{1, 2, \dots, n\}$ występuje dokładnie raz oraz w każdej kolumnie co najwyżej raz.

Czy każdy prostokąt łaciński o $m < n$ wierszach można rozszerzyć o jeden wiersz?

Wskazówka: przydatne mogą okazać się skojarzenia.

11. Krawędzie pewnego grafu spójnego G niezawierającego pętli ani krawędzi równoległych pokolorowano na czerwono, zielono i niebiesko. G ma przynajmniej 4 krawędzie oraz przynajmniej jedną krawędź każdego z trzech kolorów. Czy graf ten w każdym przypadku zawiera drzewo rozpinające trójkolorowe?

A gdyby kolorów było cztery i G był dodatkowo dwudzielny, czy zawsze zawierałby drzewo rozpinające zawierające przynajmniej jedną krawędź każdego z czterech kolorów? Czy dwudzielność jest potrzebna?

12. Organizowany jest turniej n osób, w którym każdy gra z każdym. Każda rozgrywka kończy się wygraną dokładnie jednej z osób - nie ma remisów. Wynik turnieju to graf pełny skierowany na n wierzchołkach, w którym krawędź skierowana z u do v oznacza wygraną wierzchołka u w meczu z v . Czy możliwy jest wynik turnieju, w którym różnica liczby wygranych dowolnych dwóch osób jest mniejsza od 1?

Ogólnie, czy dla każdego ciągu n liczb całkowitych dodatnich a_1, a_2, \dots, a_n takiego, że $\sum_{i=1}^n a_i = \binom{n}{2}$ istnieje wynik turnieju taki, że osoba i wygrała dokładnie a_i pojedynków?

W obu przypadkach pokaż algorytm znajdowania takiego rozkładu, o ile istnieje.

Wskazówka: przydatne mogą okazać się przepływy.

Zadania 1 – 6 są za 2 punkty, natomiast zadania 7 – 12 za 4.

Powodzenia!

Egzamin z matematyki dyskretnej

11 lutego 2020

- Niech $G = (V, E)$ oznacza graf, w którym $V = \{a_1, a_2, \dots, a_n, b_1, b_2, \dots, b_n\}$, wierzchołki a_1, a_2, \dots, a_n są połączone w cykl (tzn. dla każdego i , $1 \leq i < n$ wierzchołki a_i i a_{i+1} są połączone krawędzią oraz krawędziami są połączone a_1 i a_n), wierzchołki b_1, b_2, \dots, b_n również są połączone w cykl oraz każdy wierzchołek a_i jest połączony z każdym wierzchołkiem b_j .

Znajdź optymalne kolorowanie wierzchołkowe G . Potrzebne jest uzasadnienie.

- Wykaż, że liczba $53^{33} - 33^{33}$ jest podzielna przez 10.
- Znajdź ogólną postać rozwiązania następującego równania rekurencyjnego za pomocą anihilatorów:

$$a_{n+2} = 5a_{n+1} - 6a_n + \frac{\binom{n}{2}}{2^n}, \text{ gdy } a_0 = a_1 = 0.$$

- Podaj funkcję tworzącą dla ciągu $a_n = 1 + 2 + \dots + 2^n + (-\sqrt{2})^n$.
- W pewnej grupie muzykujących osób jedna gra na fortepianie, harfie i skrzypcach, druga na kontrabasie, harfie i wiolonczeli, trzecia na skrzypcach, czwarta na wiolonczeli i piąta na skrzypcach i wiolonczeli.

Chcieliby zagrać utwór na fortepian, skrzypce, wiolonczelę, kontrabas i harfę. Czy uda im się dobrze skłądzić?

- Niech G będzie grafem spójnym o m krawędziach. Dla każdej krawędzi e tego grafu mamy zadaną liczbę p_e oznaczającą wymaganą liczbę przejść tą krawędzią. Opracuj algorytm, który albo orzeka istnienie trasy w grafie G , w której każda krawędź jest strawersowana dokładnie p_e razy, albo stwierdza, że taka trasa nie istnieje. Punkt startu trasy nie musi być taki sam jak mety.

- Z macierzy $n \times n$ usuwamy część nad przekątną otrzymując macierz "schodkową". Na ile sposobów można ją podzielić na n prostokątów?
- Pewną grupę 24 osób składającą się z 12 kibiców drużyny A i 12 drużyn B chcemy rozmielić w czterech przedziałach 6-osobowych o numerach 1–4. Rozmieszczenia wewnętrz przedziałów są nieistotne.

- Na ile sposobów możemy to zrobić, aby w żadnym przedziale nie było tyle samo kibiców drużyny A co kibiców drużyny B ?
- Ile jest rozmieszczeń, w których sumaryczna liczba w przedziałach 1 i 2 kibiców A nie jest taka sama jak kibiców B ?

- Danych jest 12 różnych liczb dwucyfrowych. Wykaż, że wśród nich istnieją takie dwie, których różnica jest liczbą dwucyfrową o jednakowych cyfrach.

10. *Kwadratem łacińskim* nazywamy kwadrat $n \times n$, w którym na każdym polu stoi liczba ze zbioru $\{1, 2, \dots, n\}$ tak, że w każdej kolumnie oraz w każdym wierszu jest po jednej z liczb $\{1, 2, \dots, n\}$. *Prostokątem łacińskim* nazywamy prostokąt o n kolumnach i m wierszach, $1 \leq m \leq n$, w którym na każdym polu stoi liczba ze zbioru $\{1, 2, \dots, n\}$ tak, że w każdym wierszu każda z liczb $\{1, 2, \dots, n\}$ występuje dokładnie raz oraz w każdej kolumnie co najwyżej raz.

Czy każdy prostokąt łaciński o $m < n$ wierszach można rozszerzyć o jeden wiersz?

Wskazówka: przydatne mogą okazać się skojarzenia.

11. Krawędzie pewnego grafu spójnego G niezawierającego pętli ani krawędzi równoległych pokolorowano na czerwono, zielono i niebiesko. G ma przynajmniej 4 krawędzie oraz przynajmniej jedną krawędź każdego z trzech kolorów. Czy graf ten w każdym przypadku zawiera drzewo rozpinające trójkolorowe?

A gdyby kolorów było cztery i G był dodatkowo dwudzielny, czy zawsze zawierałby drzewo rozpinające zawierające przynajmniej jedną krawędź każdego z czterech kolorów? Czy dwudzielność jest potrzebna?

12. Organizowany jest turniej n osób, w którym każdy gra z każdym. Każda rozgrywka kończy się wygraną dokładnie jednej z osób - nie ma remisów. Wynik turnieju to graf pełny skierowany na n wierzchołkach, w którym krawędź skierowana z u do v oznacza wygraną wierzchołka u w meczu z v . Czy możliwy jest wynik turnieju, w którym różnica liczby wygranych dowolnych dwóch osób jest mniejsza od 1?

Ogólnie, czy dla każdego ciągu n liczb całkowitych dodatnich a_1, a_2, \dots, a_n takiego, że $\sum_{i=1}^n a_i = \binom{n}{2}$ istnieje wynik turnieju taki, że osoba i wygrała dokładnie a_i pojedynków?

W obu przypadkach pokaż algorytm znajdowania takiego rozkładu, o ile istnieje.

Wskazówka: przydatne mogą okazać się przepływy.

Zadania 1 – 6 są za 2 punkty, natomiast zadania 7 – 12 za 4.

Powodzenia!

Egzamin z matematyki dyskretnej

25 lutego 2020

- Z kliki n -wierzchołkowej K_n (klika to graf pełny, w którym każde dwa wierzchołki połączone są krawędzią) usunięto (a) jedną krawędź, (b) dwie niesąsiadujące krawędzie (bez wspólnego wierzchołka). Znajdź optymalne kolorowanie wierzchołkowe tak otrzymanego grafu. Potrzebne jest uzasadnienie.

- Znajdź ogólną postać rozwiązania następującego równania rekurencyjnego za pomocą anihilatorów:

$$a_{n+2} = 9a_n + 7\binom{n}{2}(-3)^n, \text{ gdy } a_0 = a_1 = 1.$$

- Podaj funkcję tworzącą dla ciągu: $a_n = \frac{i}{3^i}$, jeśli n jest parzyste oraz $a_n = 0$, jeśli n jest nieparzyste.

- Ile jest pokolorowań pól szachownicy 3×3 kolorami niebieskim i czerwonym takich, że nie ma kwadratów 2×2 pokolorowanych na czerwono?

- Pokaż, że w każdym grafie prostym (bez pętli i krawędzi równoległych) istnieją dwa wierzchołki o takim samym stopniu.

- W pewnej grupie muzykujących osób Ania gra na skrzypcach, harfie, kontrabasie i wiolonczeli, Basia gra na harfie i fortepianie, Cezary gra na fortepianie, Franek gra na fortepianie, Dąbrówka gra na harfie i Elwira gra na kontrabasie, skrzypcach, wiolonczeli i harfie.

Chcieliby zagrać utwór na fortepian, skrzypce, wiolonczelę, kontrabas i harfę. Czy uda im się dobrze skłądzić?

- W pewnym turnieju wzięło udział n drużyn ($n \geq 3$). Każda drużyna rozegrała z każdą dokładnie jeden mecz i nie zanotowano remisów. Udowodnij, że jeżeli pewne dwie drużyny wygrały tę samą ilość meczów, to znalazły się takie trzy drużyny A, B, C , że A wygrała z B , B wygrała z C i C wygrała z A .

- Pewien człowiek ma 7 przyjaciół. Określ, na ile sposobów może zapraszać po 3 z nich na kolacje przez 7 kolejnych dni tak, aby każdy z nich został zaproszony co najmniej raz.

- Lazik kwadratowy* jest umieszczony w punkcie $(0, 0)$ dwuwymiarowej przestrzeni euklidesowej. W dowolnym ruchu przesuwa się dokładnie o 1 w prawo, lewo, w górę lub w dół. Ile różnych tras może pokonać używając dokładnie n ruchów poziomych (prawo lub lewo) i n ruchów pionowych (w górę lub w dół), jeśli na zakończenie ma się znów znaleźć w punkcie $(0, 0)$? Kolejność robi różnicę, czyli trasy P, D, L, G oraz D, P, G, L są traktowane jako różne.

- Talią* grafu G nazywamy długość najkrótszego cyklu G . Pokaż, że graf k -regularny (taki, w którym stopień każdego wierzchołka wynosi k) o taliach co najmniej 4 ma co najmniej $2k$ wierzchołków.

11. Niech $\chi(G)$ oznacza liczbę chromatyczną grafu G (najmniejszą liczbę kolorów, jaką można pokolorować wierzchołkowo graf), a $n(G)$ liczbę wierzchołków grafu G . Przez \bar{G} oznaczamy dopełnienie grafu G (czyli graf, który zawiera krawędź $\{u, v\}$ wtedy, gdy G jej nie zawiera). Pokaż, że $\chi(G) + \chi(\bar{G}) \leq n(G) + 1$.

Wskazówka: użyj indukcji.

12. Organizowany jest turniej n osób, w którym każdy gra z każdym. Każda rozgrywka kończy się wygraną dokładnie jednej z osób - nie ma remisów. Wynik turnieju to graf pełny skierowany na n wierzchołkach, w którym krawędź skierowana z u do v oznacza wygraną wierzchołka u w meczu z v . Czy możliwy jest wynik turnieju, w którym różnica liczby wygranych dowolnych dwóch osób jest niewiększa od 1?

Ogólnie, czy dla każdego ciągu n liczb całkowitych dodatnich a_1, a_2, \dots, a_n takiego, że $\sum_{i=1}^n a_i = \binom{n}{2}$ istnieje wynik turnieju taki, że osoba i wygrała dokładnie a_i pojedynków?

W obu przypadkach pokaż algorytm znajdowania takiego rozkładu, o ile istnieje.

Wskazówka: przydatne mogą okazać się przepływy. Dodaj po jednym wierzchołku dla każdej osoby i po jednym dla każdej pary osób - w sumie $n + \binom{n}{2}$ wierzchołków.

Zadania 1 – 6 są za 2 punkty, natomiast zadania 7 – 12 za 4.

Powodzenia!

Matematyka dyskretna (L)

Katarzyna Paluch

Instytut Informatyki, Uniwersytet Wrocławski

2021

Iloczyn kartezjański

Ala chce kupić po jednej paczce: herbaty, kawy i ciastek. W sklepie, w którym robi zakupy jest 5 rodzajów herbaty, 10 rodzajów kawy i 5 typów ciastek. Na ile sposobów może skomponować swój zestaw?

Iloczyn kartezjański

Ala chce kupić po jednej paczce: herbaty, kawy i ciastek. W sklepie, w którym robi zakupy jest 5 rodzajów herbaty, 10 rodzajów kawy i 5 typów ciastek. Na ile sposobów może skomponować swój zestaw?

$$5 \cdot 10 \cdot 5$$

Iloczyn kartezjański

Niech A_1, A_2, \dots, A_n będą skończonymi zbiorami. Wówczas $|A_1 \times A_2 \times \dots \times A_n| = |A_1| \times |A_2| \times \dots \times |A_n|$.

Wariacje

Profesor Ksawery Ksenofiliński chciałby wysłać po widokówce do każdego z 7 swoich przyjaciół. Na ile sposobów może to zrobić, jeśli widokówek na ulicznym straganie jest 13 rodzajów?

Wariacje

Profesor Ksawery Ksenofiliński chciałby wysłać po widokówce do każdego z 7 swoich przyjaciół. Na ile sposobów może to zrobić, jeśli widokówek na ulicznym straganie jest 13 rodzajów?

$$13^7$$

Liczba funkcji

Liczba wariacji z powtórzeniami

Niech A i B będą skończonymi zbiorami o odpowiednio m i n elementach.
Wówczas liczba funkcji ze zbioru A w B wynosi n^m .

Innymi słowy: $|\{f : A \rightarrow B\}| = n^m$.

Liczba funkcji

Liczba wariacji z powtórzeniami

Niech A i B będą skończonymi zbiorami o odpowiednio m i n elementach.
Wówczas liczba funkcji ze zbioru A w B wynosi n^m .
Innymi słowy: $|\{f : A \rightarrow B\}| = n^m$.

Dowód 1: przez indukcję.

Dowód 2: pokazujemy równoliczność zbiorów: (i) $\{f : A \rightarrow B\}$ oraz (ii) iloczynu kartezjańskiego $B \times B \times \dots \times B$.

Wariacje cd

Profesor Ksawery Ksenofiliński wybiera się na tygodniowy rejs po Cykladach. Każdego dnia chciałby wysłać po jednej widokówce do każdego z 7 swoich przyjaciół. Okazuje się, że każdego dnia na każdej z odwiedzonych 7 (różnych) wysp sprzedawca ma 13 rodzajów widokówek do zaoferowania. Na ile sposobów profesor może wysłać widokówki?

Wariacje bez powtórzeń

Profesor Ksawery Ksenofiliński chciałby wysłać po widokówce do każdego z 7 swoich przyjaciół. Na ile sposobów może to zrobić, jeśli widokówek na ulicznym straganie jest 13 rodzajów, ale straganiarz wyprzedał prawie wszystkie widokówki i **z każdego rodzaju została tylko jedna ?**

Wariacje bez powtórzeń

Profesor Ksawery Ksenofiliński chciałby wysłać po widokówce do każdego z 7 swoich przyjaciół. Na ile sposobów może to zrobić, jeśli widokówek na ulicznym straganie jest 13 rodzajów, ale straganiarz wyprzedał prawie wszystkie widokówki i **z każdego rodzaju została tylko jedna ?**

$$13 \cdot 12 \cdot 11 \cdot \dots \cdot 7$$

Liczba funkcji różnowartościowych

Liczba wariacji bez powtórzeń

Niech A i B będą skończonymi zbiorami o odpowiednio m i n elementach. Wówczas liczba funkcji *różnowartościowych* ze zbioru A w B wynosi $n(n - 1) \dots (n - m + 1) = \frac{n!}{(n-m)!}$.

Liczba podzbiorów

Niech A będzie skończonym zbiorem o n elementach - $|A| = n$.
Ile podzbiorów ma A ?
 $|\{B : B \subseteq A\}| = ???$

Liczba podzbiorów

Liczba podzbiorów

Niech A będzie skończonym zbiorem o n elementach. Wtedy $|\{B : B \subseteq A\}| = 2^n$.

Dowód 1: przez indukcję.

Dowód 2:

Liczba podzbiorów

Liczba podzbiorów

Niech A będzie skończonym zbiorem o n elementach. Wtedy $|\{B : B \subseteq A\}| = 2^n$.

Dowód 1: przez indukcję.

Dowód 2: przez pokazanie równoliczności zbiorów: $\{B : B \subseteq A\}$ i $\{f : A \rightarrow \{0, 1\}\}$.

Para podzbiorów

Niech U będzie zbiorem n -elementowym. Na ile sposobów możemy wybrać dwa jego podzbiory A i B takie, że $A \subseteq B$?

$$|\{(A, B) : A \subseteq B \subseteq U\}| = ???$$

Para podzbiorów

Niech U będzie zbiorem n -elementowym. Na ile sposobów możemy wybrać dwa jego podzbiory A i B takie, że $A \subseteq B$?

$$|\{(A, B) : A \subseteq B \subseteq U\}| = |\{f : U \rightarrow \{0, 1, 2\}\}| = 3^n$$

Permutacje

Niech U będzie zbiorem n -elementowym. Na ile sposobów możemy ustawić w rząd jego elementy?

Permutacje

Niech U będzie zbiorem n -elementowym. Na ile sposobów możemy ustawić w rząd jego elementy?

Na tyle, ile jest funkcji różnowartościowych $f : U \rightarrow \{1, 2, \dots, n\}$.

$$|\{f : U \rightarrow \{1, 2, \dots, n\}, f \text{ różnowartościowa}\}| = \frac{n!}{(n-n)!} = n!.$$

Sufit i podłoga

Niech $x \in R$ i $n \in Z$.

$\lfloor x \rfloor = n \Leftrightarrow n \leq x < n + 1$ podłoga z x

$\lceil x \rceil = n \Leftrightarrow n - 1 < x \leq n$ sufit z x

$\{x\} = x - \lfloor x \rfloor$ część ułamkowa x

Sufit i podłoga - własności

Niech $x \in R$ i $n \in Z$.

$$\lfloor x + n \rfloor = n + \lfloor x \rfloor, \text{ bo}$$

$$\lfloor x \rfloor + n \leq x + n < \lfloor x \rfloor + n + 1$$

$$\lceil x + n \rceil = n + \lceil x \rceil$$

Sufit i podłoga - własności

Niech $x \in R$ i $n \in Z$.

Czy zachodzi: $\lfloor nx \rfloor = n\lfloor x \rfloor$?

Jak zamienić podłogę na sufit?

Sufit i podłoga - własności

Niech $x \in R$ i $n \in Z$.

Czy zachodzi: $\lfloor nx \rfloor = n\lfloor x \rfloor$?

Jak zamienić podłogę na sufit?

$$\lfloor -x \rfloor = -\lceil x \rceil$$

Matematyka dyskretna (L)

Katarzyna Paluch

Instytut Informatyki, Uniwersytet Wrocławski

2020

Podzbiory k -elementowe

Ille jest k -elementowych podzbiorów zbioru n -elementowego?

Podzbiory k -elementowe

Ille jest k -elementowych podzbiorów zbioru n -elementowego?

$$|U| = \{1, 2, \dots, n\}$$

$$P_n^k = \{A \subseteq U : |A| = k\}$$

Porównajmy P_n^k z wariacjami k -elementowymi bez powtórzeń.

$$|D| = \{1, 2, \dots, k\}$$

$$F_{k,n}^{1-1} = \{f : D \rightarrow U : f \text{ różnowartościowa } \}$$

Podzbiory k -elementowe

Ille jest k -elementowych podzbiorów zbioru n -elementowego?

$$|U| = \{1, 2, \dots, n\}$$

$$P_n^k = \{A \subseteq U : |A| = k\}$$

Porównajmy P_n^k z wariacjami k -elementowymi bez powtórzeń.

$$|D| = \{1, 2, \dots, k\}$$

$$F_{k,n}^{1-1} = \{f : D \rightarrow U : f \text{ różnowartościowa}\}$$

Dla $k = 1$ zachodzi: $|F_{k,n}^{1-1}| = |P_n^k|$

Dla $k > 1$ zachodzi: $|F_{k,n}^{1-1}| > |P_n^k|$

Podzbiory k -elementowe

Ille jest k -elementowych podzbiorów zbioru n -elementowego?

- Elementy k -elementowego podzbioru U możemy ustawić na $k!$ sposobów.
- Każdemu k -elem. podzbiorowi A odpowiada $k!$ funkcji różnowartościowych $\{1, 2, \dots, k\} \rightarrow A$.
- Każdemu k -elem. podzbiorowi A odpowiada $k!$ -elem zbiór Z_A .
- Zauważmy, że $A \neq B \Rightarrow Z_A \cap Z_B = \emptyset$.
- $F_{k,n}^{1-1} = \bigcup_{A \subseteq U, |A|=k} Z_A$
- $|F_{k,n}^{1-1}| = k!|P_n^k|$
- $\frac{n!}{(n-k)!} = k!|P_n^k|$
- $|P_n^k| = \frac{n!}{(n-k)!k!} = \binom{n}{k}$

Symbol Newtona - własności

Niech $k, n \in N$ takie, że $0 \leq k \leq n$.

Wówczas $\binom{n}{k} = \binom{n}{n-k}$

Symbol Newtona - własności

Niech $k, n \in N$ takie, że $0 \leq k \leq n$.

Wówczas $\binom{n}{k} = \binom{n}{n-k}$

Dowód 1: algebraiczny.

Dowód 2 kombinatoryczny: budujemy bijekcję \mathcal{F} między P_n^k i P_n^{n-k} .

$$P_n^k = \{A \subseteq \{1, 2, \dots, n\} : |A| = k\}$$

$$P_n^{n-k} = \{A \subseteq \{1, 2, \dots, n\} : |A| = n - k\}$$

Symbol Newtona - własności

Niech $k, n \in N$ takie, że $0 \leq k \leq n$.

Wówczas $\binom{n}{k} = \binom{n}{n-k}$

Dowód 1: algebraiczny.

Dowód 2 kombinatoryczny: budujemy bijekcję \mathcal{F} między P_n^k i P_n^{n-k} .

$$P_n^k = \{A \subseteq \{1, 2, \dots, n\} : |A| = k\}$$

$$P_n^{n-k} = \{A \subseteq \{1, 2, \dots, n\} : |A| = n - k\}$$

$$\mathcal{F} : P_n^k \rightarrow P_n^{n-k}$$

$$\mathcal{F}(A) = \{1, 2, \dots, n\} \setminus A = \bar{A} \text{ (A przyporządkowujemy dopełnienie } A)$$

Symbol Newtona - własności

Niech $k, n \in N$ takie, że $0 \leq k < n$.

$$\text{Wówczas } \binom{n}{k} + \binom{n}{k+1} = \binom{n+1}{k+1}$$

Dowód 1: algebraiczny.

Dowód 2 kombinatoryczny:

Symbol Newtona - własności

Niech $k, n \in N$ takie, że $0 \leq k < n$.

Wówczas $\binom{n}{k} + \binom{n}{k+1} = \binom{n+1}{k+1}$

Dowód 1: algebraiczny.

Dowód 2 kombinatoryczny: dzielimy P_{n+1}^{k+1} na dwa rozłączne zbiory:

$$U = \{1, 2, \dots, n+1\}$$

Z_+^{k+1} - zbiór $(k+1)$ -elem. podzbiorów U **zawierających $n+1$**

Z_-^{k+1} - zbiór $(k+1)$ -elem. podzbiorów U **niezawierających $n+1$**

Symbol Newtona - własności

Niech $k, n \in N$ takie, że $0 \leq k < n$.

$$\text{Wówczas } \binom{n}{k} + \binom{n}{k+1} = \binom{n+1}{k+1}$$

Dowód 1: algebraiczny.

Dowód 2 kombinatoryczny: dzielimy P_{n+1}^{k+1} na dwa rozłączne zbiory:

$$U = \{1, 2, \dots, n+1\}$$

Z_+^{k+1} - zbiór $(k+1)$ -elem. podzbiorów U **zawierających $n+1$**

Z_-^{k+1} - zbiór $(k+1)$ -elem. podzbiorów U **niezawierających $n+1$**

$$|P_{n+1}^{k+1}| = |Z_+^{k+1}| + |Z_-^{k+1}|$$

$$|Z_-^{k+1}| = |P_n^{k+1}| = \binom{n}{k+1}$$

Symbol Newtona - własności

Niech $k, n \in N$ takie, że $0 \leq k < n$.

$$\text{Wówczas } \binom{n}{k} + \binom{n}{k+1} = \binom{n+1}{k+1}$$

Dowód 1: algebraiczny.

Dowód 2 kombinatoryczny: dzielimy P_{n+1}^{k+1} na dwa rozłączne zbiory:

$$U = \{1, 2, \dots, n+1\}$$

Z_+^{k+1} - zbiór $(k+1)$ -elem. podzbiorów U **zawierających $n+1$**

Z_-^{k+1} - zbiór $(k+1)$ -elem. podzbiorów U **niezawierających $n+1$**

$$|P_{n+1}^{k+1}| = |Z_+^{k+1}| + |Z_-^{k+1}|$$

$$|Z_-^{k+1}| = |P_n^{k+1}| = \binom{n}{k+1}$$

$$|Z_+^{k+1}| = |P_n^k| = \binom{n}{k}$$

Trójkąt Pascala

			1		
	1		1		
1		2	1		
1		3	3	1	

Kulki i szufladki

Na ile sposobów można wrzucić n (nieroóżnialnych) kulek do k (roróżnialnych) szuflad?

Kulki i szufladki

Na ile sposobów można wrzucić n kulek do k szuflad?

Zakodujmy każdy rozrzuć za pomocą zer i jedynek, tzn. jako ciąg zerojedynkowy.

Kulki i szufladki

Na ile sposobów można wrzucić n kulek do k szuflad?

Zakodujmy każdy rozrzuć za pomocą zer i jedynek, tzn. jako ciąg zerojedynkowy.

Użyjemy n zer - reprezentują kulki i $k - 1$ jedynek, które są oddzielaczami.

Interpretacja: ilość zer między $(i - 1)$ -szą i i -tą jedynką to ilość kulek w i -tej szufladzie.

Przykład: 0011000 oznacza 2-kulki w pierwszej, 0 kulki w drugiej, 3 kulki w trzeciej.

Kulki i szufladki

Na ile sposobów można wrzucić n kulek do k szuflad?

Na tyle, ile jest ciągów złożonych z n zer i $k - 1$ jedynek.

Każdy taki ciąg ma długość $n + k - 1$.

Trzeba wybrać $k - 1$ miejsc spośród $n + k - 1$, na których postawimy jedynkę.

Odpowiedź: $\binom{n+k-1}{k-1}$

Wzór dwumienny Newtona

Dla $n \in N$ zachodzi:

$$(x + y)^n = \sum_{i=0}^n \binom{n}{i} x^i y^{n-i}$$

Dwumian Newtona

Wzór dwumienny Newtona

Dla $n \in N$ zachodzi:

$$(x + y)^n = \sum_{i=0}^n \binom{n}{i} x^i y^{n-i}$$

Dowód kombinatoryczny:

$$(x + y)^n = \sum_{i=0}^n \alpha_i x^i y^{n-i}$$

$$(x + y)^n = (x + y)(x + y) \cdots (x + y) = x^n + y^n + \sum_{i=1}^{n-1} \alpha_i x^i y^{n-i}$$

Mamy 2^n mnożeń.

α_i to liczba sposobów, na jakie możemy wybrać i spośród n nawiasów, w których w mnożeniu uczestniczyć będzie x (a nie y)

Funkcja duże O

Niech $f, g : N \rightarrow R \geq 0$.

$$f(n) = O(g(n)) \Leftrightarrow \exists_{c>0} \exists_{n_0 \in N} \forall_{n \geq n_0} f(n) \leq cg(n)$$

Asymptotyka

Funkcja duże O

Niech $f, g : N \rightarrow R \geq 0$.

$$f(n) = O(g(n)) \Leftrightarrow \exists_{c>0} \exists_{n_0 \in N} \forall_{n \geq n_0} f(n) \leq cg(n)$$

- Czy $f(n) = O(f(n))$?
- Czy $f(n) = O(\frac{f(n)}{100})$?
- Czy $10^6 n = O(n^2)$?

Asymptotyka

Funkcja duże O

Niech $f, g : N \rightarrow R \geq 0$.

$$f(n) = O(g(n)) \Leftrightarrow \exists_{c>0} \exists_{n_0 \in N} \forall_{n \geq n_0} f(n) \leq cg(n)$$

- Czy $f(n) = O(f(n))?$ $n_0 = 0, c = 1$
- Czy $f(n) = O(\frac{f(n)}{100})?$ $n_0 = 0, c = 100$
- Czy $10^6 n = O(n^2)?$ $n_0 = 10^6, c = 1$

Asymptotyka

Funkcja małe o

Niech $f, g : N \rightarrow R \geq 0$.

$$f(n) = o(g(n)) \Leftrightarrow \lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = 0$$

Asymptotyka

Funkcja małe o

Niech $f, g : N \rightarrow R \geq 0$.

$$f(n) = o(g(n)) \Leftrightarrow \lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = 0$$

- Czy $f(n) = o(f(n))$?
- Czy $f(n) = o(g(n)) \Rightarrow f(n) = O(g(n))$?
- Czy $f(n) = o(g(n)) \Rightarrow g(n) = o(f(n))$?

Asymptotyka

Funkcja małe o

Niech $f, g : N \rightarrow R \geq 0$.

$$f(n) = o(g(n)) \Leftrightarrow \lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = 0$$

- Czy $f(n) = o(f(n))$? **NIE**
- Czy $f(n) = o(g(n)) \Rightarrow f(n) = O(g(n))$? **TAK**
- Czy $f(n) = o(g(n)) \Rightarrow g(n) = o(f(n))$? **NIE**

Duże O

Niech $C, a, \alpha, \beta \in R > 0$.

- $\forall_{\alpha, \beta} \alpha \leq \beta \Rightarrow n^\alpha = O(n^\beta)$
- $\forall_{a>1} n^C = O(a^n)$
- $\forall_{\alpha>0} (\ln n)^C = O(n^\alpha)$

Asymptotyka

Duże O

Niech $C, a, \alpha, \beta \in R > 0$.

- $\forall_{\alpha, \beta} \alpha \leq \beta \Rightarrow n^\alpha = O(n^\beta)$
- $\forall_{a>1} n^C = O(a^n)$
- $\forall_{\alpha>0} (\ln n)^C = O(n^\alpha)$

Przydatna może się okazać reguła de l'Hospitala:

Jeśli $f(n)$ i $g(n)$ dążą do nieskończności, to $\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = \lim_{n \rightarrow \infty} \frac{f'(n)}{g'(n)}$.

Inne funkcje

Niech $f, g : N \rightarrow R \geq 0$.

- $f(n) = \Omega(g(n)) \Leftrightarrow \exists_{c>0} \exists_{n_0 \in N} \forall_{n \geq n_0} f(n) \geq cg(n)$
- $f(n) = \Theta(g(n)) \Leftrightarrow f(n) = \Omega(g(n)) \wedge f(n) = O(g(n))$
- $f(n) = \omega(g(n)) \Leftrightarrow \lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = \infty$

Suma harmoniczna

$$S_n = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n} = \sum_{i=1}^n \frac{1}{i}$$

$$G_k = \left\{ i : \frac{1}{2^k} < \frac{1}{i} \leq \frac{1}{2^{k-1}} \right\}$$

Kulki i szufladki c.d.

Na ile sposobów można wrzucić n kulek do k szufladek tak, aby żadna szuflada nie była pusta?

Matematyka dyskretna (L)

Katarzyna Paluch

Instytut Informatyki, Uniwersytet Wrocławski

2021

Szachownica i domino

Z szachownicy 8×8 wycinamy jedno pole z narożnika.

Czy tak zdeformowaną szachownicę można pokryć kostkami domino, jeśli każda taka kostka obejmuje dwa pola szachownicy?

Szachownica i domino

Z szachownicy 8×8 wycinamy dwa pola z przeciwnieległych narożników.

Czy taką szachownicę można pokryć kostkami domino?

Szachownica i pchły

W środku każdego pola szachownicy 5×5 siedzi pchła.

Na sygnał ka da z pche  przeskakuje na jakie  sasiaduj ce pole. Dwa pola s  sasiaduj ce, je li maj  wsp lny bok.

Czy istnieje strategia gwarantuj ca,  e na ka dym polu ponownie b edzie dok adnie jedna pch a?

Zasada szufladkowa Dirichleta

Zasada szufladkowa

Niech $k, s \in N > 0$.

Jeśli wrzucimy k kulek do s szuflad (Dirichleta) a kulek jest więcej niż szuflad ($k > s$), to w którejś szufladzie znajdą się przynajmniej 2 kulki.

Zasada szufladkowa Dirichleta

Zasada szufladkowa

Niech A i B będą skończonymi zbiorami.

Wówczas, jeśli $|A| > |B|$, to nie istnieje funkcja różnowartosciowa z A w B .

Zasada szufladkowa Dirichleta

Zasada szufladkowa

Niech $k, s \in \mathbb{N} > 0$.

Jeśli wrzucimy $k > s \cdot i$ kulek do s szuflad (Dirichleta), to w którejś szufladzie znajdą się przynajmniej $i + 1$ kulki.

Krzesła i ludzie

W rzędzie stoi 12 krzeseł. Zajmuje je 9 osób.

Pokaż, że w każdym przypadku jakieś 3 sąsiadujące krzesła zostaną zajęte.

Liczba znajomych

Pokaż, że w dwolnej grupie n osób ($n \in N$) znajdą się 2 osoby o takiej samej liczbie znajomych (z tej grupy).

Dwukolorowa płaszczyzna

Każdy punkt płaszczyzny kolorujemy na jeden z dwóch kolorów: szmaragdowy lub koralowy.

Pokaż, że znajdą się dwa punkty w odległości dokładnie 1 i tego samego koloru.

55 liczb

Wybieramy 55 liczb naturalnych takich, że:
 $1 \leq x_1 < x_2 < \dots < x_{55} \leq 100$.

Pokaż, że jakkolwiek byśmy je nie wybrali, jakieś dwie będą różnić się o 9.

Funkcja modulo

Niech $n, d \in \mathbb{Z}$ i $d \neq 0$.

$$n \bmod d = n - \lfloor \frac{n}{d} \rfloor d$$

$$n \bmod d = r \Leftrightarrow 0 \leq r < d \wedge \exists_{k \in \mathbb{Z}} n = kd + r$$

Funkcja modulo - własności

$$(a + b) \bmod n = (a \bmod n + b \bmod n) \bmod n$$
$$(a \cdot b) \bmod n = ((a \bmod n) \cdot (b \bmod n)) \bmod n$$

Przystawanie modulo:

$$a \equiv_n b \Leftrightarrow a \bmod n = b \bmod n$$

$$a + b \equiv_n a \bmod n + b \bmod n$$

$$a \cdot b \equiv_n (a \bmod n) \cdot (b \bmod n)$$

Podzielność

Niech $n, d \in \mathbb{Z}$ i $d \neq 0$.

$$d|n \Leftrightarrow \exists_{k \in \mathbb{Z}} n = kd$$

$$d|n \Leftrightarrow n \bmod d = 0$$

$$d|n \Leftrightarrow n \equiv_d 0$$

Podzielność- własności

$$d|n_1 \wedge d|n_2 \Rightarrow d|(n_1 + n_2)$$

Czy zachodzi implikacja w drugą stronę?

Podzielność przez 7

Pokaż, że wśród dowolnych 8 liczb całkowitych różnica jakichś dwóch dzieli się przez 7.

Potęgi 3

Pokaż, że istnieją dwie potęgi 3, których różnica dzieli się przez 2021.

Podzielność przez 3

Podzielność przez 3

Liczba naturalna x dzieli się przez 3 wtedy, gdy suma jej cyfr w zapisie dziesiętnym dzieli się przez 3.

NWD

Niech $a, b \in N$.

$$NWD(a, b) = \max\{d \in N : d|a \wedge d|b\}$$

Algorytm Euklidesa

Algorytm Euklidesa

$a \geq b > 0$

$NWD(a, b) = NWD(b, a \bmod b)$

$NWD(a, 0) = a$

Rozszerzony algorytm Euklidesa

Rozszerzony algorytm Euklidesa

$$a \geq b > 0$$

$$\exists_{x,y \in \mathbb{Z}} \quad xa + yb = NWD(a, b)$$

Rozszerzony algorytm Euklidesa

$$xa + yb = NWD(a, b)$$

$$x'b + y'(a \bmod b) = NWD(b, a \bmod b) = NWD(a, b)$$

$$x'b + y'(a - b\lfloor \frac{a}{b} \rfloor) = NWD(a, b)$$

$$y'a + (x' - \lfloor \frac{a}{b} \rfloor)b = NWD(a, b)$$

$$x \leftarrow y', \quad y \leftarrow x' - \lfloor \frac{a}{b} \rfloor$$

Rozszerzony algorytm Euklidesa

Niech $a, b, d \in N$, $x, y \in Z$ oraz $a > 0$.

$$d|a \wedge d|b \wedge xa + yb = d$$

Czy to znaczy, że $d = NWD(a, b)$?

Liczby względnie pierwsze

Niech $a, b \in \mathbb{Z}$.

a i b są względnie pierwsze gdy $NWD(a, b) = 1$.

Matematyka dyskretna (L)

Katarzyna Paluch

Instytut Informatyki, Uniwersytet Wrocławski

2020

Sporty rakietowe

W pewnym klubie każdy gra w squasha lub badmintona.

- 25 osób gra w badmintona,
- 40 w squasha,
- 10 gra i w badmintonie, i w squasha.

Ille osób jest w tym klubie?

Sporty rakietowe

W pewnym klubie każdy gra w squasha lub badmintona, lub w tenisa.

- 25 osób gra w badmintona,
- 40 w squasha,
- 10 gra i w badmintona, i w squasha,
- 30 gra w tenisa,
- 5 w tenisa i badmintona,
- 7 w tenisa i squasha,
- 3 we wszystkie trzy sporty.

Ille osób jest w tym klubie?

Sporty rakietowe

W pewnym klubie każdy gra w squasha lub badmintona, lub w tenisa, lub w ping-ponga.

- 25 osób gra w badmintona,
- 40 w squasha,
- 10 gra i w badmintona, i w squasha,
- 30 gra w tenisa,
- 5 w tenisa i badmintona,
- 7 w tenisa i squasha,
- 3 w tenisa, badmintona i squasha,
-

Ille osób jest w tym klubie?

Wzór włączeń i wyłączeń

$$|\bigcup_{i=1}^n A_i| = \sum_{k=1}^n (-1)^{k-1} \sum_{\emptyset \neq I \subseteq \{1, 2, \dots, n\}, |I|=k} |\bigcap_{i \in I} A_i|$$

$$|\bigcup_{i=1}^n A_i| = \sum_{\emptyset \neq I \subseteq \{1, 2, \dots, n\}} (-1)^{|I|-1} |\bigcap_{i \in I} A_i|$$

Szczególne sito

Ille liczb spośród $\{1, 2, \dots, 100\}$ nie jest podzielnych przez żadną z liczb 6, 8, 15?

Równanie

Ille rozwiązań wśród liczb naturalnych ma równanie $x_1 + x_2 + x_3 + x_4 = 50$, jeśli wymagamy, aby $x_i < 15$ dla każdego $1 \leq i \leq 4$?

Surjekcje

Ille jest różnych surjekcji z n -elem. zbioru A na m -elem. zbiór B ?

Liczby Fibonacciego

- $F_0 = 0$,
- $F_1 = 1$,
- $F_n = F_{n-1} + F_{n-2}$ dla $n > 1$.

Proste na płaszczyźnie

Na ile maksymalnie obszarów można podzielić płaszczyznę przy pomocy n prostych?

Wieża Hanoi

Ille potrzeba ruchów by przenieść wieżę składającą się z n krążków z preta A na preć C używając pomocniczo preta B ,
jeśli:

- krążki są różnej wielkości, ułożone od największego do najmniejszego, największy na spodzie, najmniejszy na wierzchu,
- w jednym ruchu można przenieść jeden krążek,
- nie można klaść krążka większego na mniejszym?

Matematyka dyskretna (L)

Katarzyna Paluch

Instytut Informatyki, Uniwersytet Wrocławski

2021

Operatory działające na ciągi

$$< a_n > = (a_0, a_1, a_2, \dots, a_n, \dots)$$

operator przesunięcia: $E < a_n > = < a_{n+1} > = (a_1, a_2, \dots, a_n, \dots)$

$$E^2 < a_n > = E(E < a_n >) = (a_2, a_3, \dots)$$

Operatory działające na ciągi

$$\langle a_n \rangle = (a_0, a_1, a_2, \dots, a_n, \dots)$$

$$\langle a_n \rangle + \langle b_n \rangle = \langle a_n + b_n \rangle = (a_0 + b_0, a_1 + b_1, a_2 + b_2, \dots, a_n + b_n, \dots)$$

$$c \langle a_n \rangle = \langle ca_n \rangle = (ca_0, ca_1, ca_2, \dots, ca_n, \dots)$$

Anihilator ciągu

Operator O jest **anihilatorem** ciągu $\langle a_n \rangle$, jeśli
 $O \langle a_n \rangle = \langle 0 \rangle = (0, 0, \dots, 0, \dots)$.

Operator mnożenia przez 0 jest trywialnym anihilatorem każdego ciągu,
dla którego nie jest traktowany jako anihilator.

Anihilator ciągu

Jaki operator anihiluje ciąg $\langle 1 \rangle = (1, 1, 1, \dots)$?

Anihilator ciągu

Jaki operator anihiluje ciąg $\langle 1 \rangle = (1, 1, 1, \dots)$?

$E - 1$

Anihilator ciągu

Jaki operator anihiluje ciąg $\langle \pi \rangle = (\pi, \pi, \pi, \dots)$?

Anihilator ciągu

Jaki operator anihiluje ciąg $\langle \pi \rangle = (\pi, \pi, \pi, \dots)$?

$E - 1$

Anihilator ciągu

Jaki operator anihiluje ciąg $\langle 2^n \rangle = (1, 2, 4, 8, \dots, 2^n, \dots)$?

Anihilator ciągu

Jaki operator anihiluje ciąg $\langle 2^n \rangle = (1, 2, 4, 8, \dots, 2^n, \dots)$?

E – 2

Anihilator ciągu

Jaki operator anihiluje ciąg $\langle \pi 2^n \rangle = (\pi, 2\pi, 4\pi, 8\pi, \dots, 2^n\pi, \dots)$?

Anihilator ciągu

Jakie ciągi anihiluje operator $E - 2$?

Anihilator ciągu

Jakie ciągi anihiluje operator $E - a$?

Anihilator ciągu

Jakie ciągi anihiluje operator $E - a$?

$\langle \alpha a^n \rangle$ dla dowolnego $\alpha \in R$

Anihilator ciągu

Jaki operator anihiluje ciąg $\langle 2^n + 3^n \rangle = (2, 5, 13, \dots, 2^n + 3^n, \dots)$?

Anihilator ciągu

Jaki operator anihiluje ciąg $\langle 2^n + 3^n \rangle = (2, 5, 13, \dots, 2^n + 3^n, \dots)$?

Operator $E - 2$ właściwie nie zmienia ciągu 3^n , tzn
 $(E - 2) \langle 3^n \rangle = \langle \alpha 3^n \rangle$ dla pewnego $\alpha \neq 0$.

Anihilator ciągu

Jaki operator anihiluje ciąg $\langle 2^n + 3^n \rangle = (2, 5, 13, \dots, 2^n + 3^n, \dots)$?

$$(E - 2)(E - 3)$$

Anihilator ciągu

Jaki operator anihiluje ciąg $\langle n2^n \rangle$?

Anihilator ciągu

Jaki operator anihiluje ciąg $\langle n2^n \rangle$?

$$(E - 2)^2$$

Anihilator ciągu

Jakie ciągi anihiluje operator $(E - 2)^2$?

Anihilator ciągu

Jakie ciągi anihiluje operator $(E - 2)^2$?

$$<(\alpha n + \beta)2^n>$$

Anihilator ciągu

Jakie ciągi anihiluje operator $(E - 2)^k$?

Anihilator ciągu

Jakie ciągi anihiluje operator $(E - 2)^k$?

$$<(\alpha_{k-1}n^{k-1} + \alpha_{k-2}n^{k-2} + \dots + \alpha_1n + \alpha_0)2^n> = <2^n \sum_{i=0}^{k-1} \alpha_i n^i>$$

Anihilator ciągu

Jakie ciągi anihiluje operator $(E - c)^k$?

Anihilator ciągu

Jakie ciągi anihiluje operator $(E - c)^k$?

$$<(\alpha_{k-1}n^{k-1} + \alpha_{k-2}n^{k-2} + \dots + \alpha_1n + \alpha_0)c^n> = <c^n \sum_{i=0}^{k-1} \alpha_i n^i>$$

Zastosowanie anihilatora

Znając anihilator ciągu $\langle a_n \rangle$, znamy $\langle a_n \rangle$.

Jeśli wiemy jaki operator **anihiluje** (zeruje/sprowadza do zera) ciąg $\langle a_n \rangle$, wiemy, jaką postać ma $\langle a_n \rangle$.

Ciąg a_n

- $a_0 = \pi$,
- $a_n = 7a_{n-1}$ dla $n \geq 1$.

Czym zanihilować $\langle a_n \rangle$?

Ciąg a_n

- $a_0 = \pi$,
- $a_n = 7a_{n-1}$ dla $n \geq 1$.

$$E < a_n > = < a_1, a_2, a_3, \dots > = < 7a_0, 7a_1, 7a_2, \dots > = 7 < a_n >$$

Ciąg a_n

- $a_0 = \pi$,
- $a_n = 7a_{n-1}$ dla $n \geq 1$.

$E < a_n > = < a_1, a_2, a_3, \dots > = < 7a_0, 7a_1, 7a_2, \dots > = 7 < a_n >$.
W takim razie $E - 7$ jest **anihilatorem** $< a_n >$.

Ciąg a_n

- $a_0 = \pi$,
- $a_n = 7a_{n-1}$ dla $n \geq 1$.

W takim razie $E - 7$ jest **anihilatorem** $\langle a_n \rangle$.

$$\langle a_n \rangle = \langle \alpha 7^n \rangle$$

Ciąg a_n

- $a_0 = \pi$,
- $a_n = 7a_{n-1}$ dla $n \geq 1$.

W takim razie $E - 7$ jest **anihilatorem** $\langle a_n \rangle$.

$$\langle a_n \rangle = \langle \alpha 7^n \rangle$$

Aby obliczyć α , rozwiązujemy równanie $\alpha 7^0 = a_0 = \pi$. Zatem $\alpha = \pi$.

Liczby Fibonacciego

- $F_0 = 0$,
- $F_1 = 1$,
- $F_n = F_{n-1} + F_{n-2}$ dla $n > 1$.

Jaki operator ahihiluje $\langle F_n \rangle$?

Liczby Fibonacciego

- $F_0 = 0,$
- $F_1 = 1,$
- $F_n = F_{n-1} + F_{n-2}$ dla $n > 1.$

Jaki operator ahihiluje $\langle F_n \rangle$?

$$E^2 \langle F_n \rangle = \langle F_{n+2} \rangle = (F_2, F_3, \dots) = (F_0 + F_1, F_1 + F_2, F_2 + F_3, \dots) = \\ (F_0, F_1, \dots) + (F_1, F_2, \dots) = \langle F_n \rangle + E \langle F_n \rangle$$

Liczby Fibonacciego

- $F_0 = 0,$
- $F_1 = 1,$
- $F_n = F_{n-1} + F_{n-2}$ dla $n > 1.$

Jaki operator ahihiluje $\langle F_n \rangle$?

$$E^2 \langle F_n \rangle = \langle F_n \rangle + E \langle F_n \rangle = (1 + E) \langle F_n \rangle$$

Zatem $E^2 - E - 1$ anihiluje $\langle F_n \rangle.$

Liczby Fibonacciego

$E^2 - E - 1$ anihiluje $\langle F_n \rangle$.

$$E^2 - E - 1 = (E - \frac{1-\sqrt{5}}{2})(E - \frac{1+\sqrt{5}}{2})$$

$$F_n = \alpha(\frac{1-\sqrt{5}}{2})^n + \beta(\frac{1+\sqrt{5}}{2})^n$$

Aby obliczyć α, β rozwiązujemy układ równań:

$$F_0 = 0 = \alpha + \beta$$

$$F_1 = 1 = \alpha \frac{1-\sqrt{5}}{2} + \beta \frac{1+\sqrt{5}}{2}$$

Ciąg a_n

- $a_0 = 0,$
- $a_1 = 1,$
- $a_n = 4a_{n-1} - 4a_{n-2} + n2^n \text{ dla } n > 1.$

Jaki jest annihilator $\langle a_n \rangle?$

Ciąg a_n

- $a_0 = 0,$
- $a_1 = 1,$
- $a_n = \sqrt{a_{n-1}^2 + a_{n-2}^2}$ dla $n > 1.$

Matematyka dyskretna (L)

Katarzyna Paluch

Instytut Informatyki, Uniwersytet Wrocławski

2021

Nawiasowanie

Na ile sposobów można w poprawny sposób ustawić n par nawiasów?
Każda para składa się z nawiasu otwierającego i zamykającego.

c_i - liczba poprawnych nawiasowań i par nawiasów

$$c_0 = 1$$

$$c_1 = 1$$

$$c_2 = 2$$

Nawiasowanie

Problem równoważny nawiasowaniu:

Założmy, że punkt startowy to $(0, 0)$ w układzie współrzędnych. Mamy do dyspozycji n ruchów \nearrow i n ruchów \searrow .

Na ile różnych sposobów możemy je wykonać tak, by nigdy nie znaleźć się pod poziomem 0?

Nawiasowanie

Jak zapisać c_i w postaci zależności rekurencyjnej?

Nawiasowanie

Jak zapisać c_i w postaci zależności rekurencyjnej?

Mожет $c_n = c_1c_{n-1} + c_2c_{n-2} \dots + c_1c_n + 1 = 1 + \sum_{i=1}^{n-1} c_i c_{n-i}$?

Nawiasowanie

Jak zapisać c_i w postaci zależności rekurencyjnej?

$$c_n = \sum_{i=1}^n d_i, \text{ gdzie}$$

d_i liczba "ponadziemnych" ustawień n par kroków \nearrow i \searrow , w której poziom 0 osiągamy po raz pierwszy po $2i$ krokach.

Nawiasowanie

$c_n = \sum_{i=1}^n d_i$, gdzie

d_i liczba "ponadziemnych" ustawień n par kroków \nearrow i \searrow , w których poziom 0 osiągamy po raz pierwszy po $2i$ krokach.

$$d_i = c_{i-1} c_{n-i}$$

Nawiasowanie

$$c_n = \sum_{i=1}^n d_i, \text{ gdzie}$$

d_i liczba "ponadziemnych" ustawień n par kroków \nearrow i \searrow , w których poziom 0 osiągamy po raz pierwszy po $2i$ krokach.

$$d_i = c_{i-1} c_{n-i}$$

$$c_n = \sum_{i=1}^n c_{i-1} c_{n-i}$$

Nawiasowanie

Problem równoważny nawiasowaniu:

Założmy, że punkt startowy to $(0, 0)$ w układzie współrzędnych. Mamy do dyspozycji n ruchów \rightarrow (przesunięcie się o 1 w prawo) i n ruchów \uparrow (przesunięcie się o 1 w górę).

Na ile różnych sposobów możemy je wykonać tak, by nigdy nie przekroczyć prostej $y = x$?

Nawiasowanie

Założymy, że punkt startowy to $(0, 0)$ w układzie współrzędnych. Mamy do dyspozycji n ruchów \rightarrow (przesunięcie się o 1 w prawo) i n ruchów \uparrow (przesunięcie się o 1 w górę).

Na ile różnych sposobów możemy je wykonać tak, by nigdy nie przekroczyć prostej $y = x$?

$$c_n = \binom{2n}{n} - \text{liczba złych ustawień}$$

Ile jest ustawień złych - przekraczających $y = x$?

Nawiasowanie

$$c_n = \binom{2n}{n} - \text{liczba złych ustnień}$$

Ile jest ustnień złych - przekraczających $y = x$?

$$c_n = \binom{2n}{n} - \binom{2n}{n-1} =$$

c_n - liczby Catalana

$$c_0 = 0, \text{ dla } n > 0 : c_n = \sum_{i=1}^n c_{i-1} c_{n-i}$$

Funkcje tworzące

Niech $\langle a_n \rangle$ będzie pewnym ciągiem.

$$\sum_{i=0}^{\infty} a_i x^i = a_0 + a_1 x + a_2 x^2 + \dots + a_i x^i + \dots$$

Jeśli $\sum_{i=0}^{\infty} a_i x^i = A(x)$ dla pewnej funkcji $A(x)$, to

$A(x)$ jest **funkcją tworzącą** ciągu $\langle a_n \rangle$.

Funkcje tworzące

Niech $\langle a_n \rangle = \langle 1 \rangle = (1, 1, 1, \dots)$.

$$\sum_{i=0}^{\infty} 1x^i = 1 + x + x^2 + \dots + x^i + \dots$$

$$\sum_{i=0}^{\infty} x^i = \frac{1}{1-x}$$

$\frac{1}{1-x}$ jest **funkcją tworzącą** ciągu $\langle 1 \rangle$.

Funkcje tworzące

Niech $\langle a_n \rangle = \langle 7 \rangle = (7, 7, 7, \dots)$.

$$\sum_{i=0}^{\infty} 7x^i = 7 + 7x + 7x^2 + \dots + 7x^i + \dots$$

$$\sum_{i=0}^{\infty} x^i = \frac{1}{1-x}$$

$\frac{7}{1-x}$ jest **funkcją tworzącą** ciągu $\langle 7 \rangle$.

Funkcje tworzące

Niech $\langle a_n \rangle = \langle 2^n \rangle = (1, 2, 4, 8, \dots)$.

$$\sum_{i=0}^{\infty} 2^i x^i = 1 + 2x + 4x^2 + \dots + 2^i x^i + \dots$$

$$\sum_{i=0}^{\infty} 2^i x^i = \sum_{i=0}^{\infty} (2x)^i = \frac{1}{1-2x}$$

$\frac{1}{1-2x}$ jest **funkcją tworzącą** ciągu $\langle 2^n \rangle$.

Funkcje tworzące

Niech $\langle a_n \rangle = \langle (-1)^n \rangle = (1, -1, 1, -1, \dots)$.

$$\sum_{i=0}^{\infty} (-1)^i x^i = 1 - x + x^2 + \dots + (-1)^i x^i + \dots$$

? jest **funkcją tworzącą** ciągu $\langle (-1)^n \rangle$.

Wydawanie reszty

Niech $\langle p_n \rangle$ = liczba sposobów na jaką można wydać kwotę n za pomocą 5-złotówek.

$p_n = 1$ dla n podzielnych przez 5, $p_n = 0$ w p.p.

$$1 + x^5 + x^{10} + x^{15} + \dots = \sum_{i=0}^{\infty} x^{5i} = \frac{1}{1-x^5}$$

Wydawanie reszty

Niech d_n = liczba sposobów na jaką można wydać kwotę n za pomocą 2-złotówek.

$d_n = 1$ dla n podzielnych przez 2, $p_n = 0$ w p.p.

$$1 + x^2 + x^4 + x^6 + \dots = \sum_{i=0}^{\infty} x^{2i} = \frac{1}{1-x^2}$$

Wydawanie reszty

Niech r_n = liczba sposobów na jaką można wydać kwotę n za pomocą 2-złotówek oraz 5-złotówek.

$$r_0 = 1, r_1 = 0, r_2 = 1, r_4 = 1, r_{10} = 2$$

Wydawanie reszty

Niech r_n = liczba sposobów na jaką można wydać kwotę n za pomocą 2-złotówek oraz 5-złotówek.

Czy funkcja tworząca dla r_n to $P(x)D(x) = \frac{1}{1-x^5} \cdot \frac{1}{1-x^2}$?

$$(1 + x^5 + x^{10} + x^{15} + \dots)(1 + x^2 + x^4 + x^6 + x^8 + \dots) = \\ 1 + x^2 + x^4 + x^5 + \dots + 2x^{10} + \dots$$

Jaki współczynnik stoi przy x^{17} ?

Równy mocy zbioru $\{x^5x^{12}, x^{15}x^2\}$.

Jaki współczynnik stoi przy x^{44} ?

Równy mocy zbioru $\{x^0x^{22}, x^{10}x^{32}, x^{20}x^{24}, x^{30}x^{14}, x^{40}x^4\}$.

Równy mocy zbioru $\{(i, j) : 5i + 2j = 44\}$.

Wydawanie reszty

Niech r_n = liczba sposobów na jaką można wydać kwotę n za pomocą 1-złotówek, 2-złotówek oraz 5-złotówek.

Czy funkcja tworząca dla r_n to $\frac{1}{1-x^5} \cdot \frac{1}{1-x^2} \cdot \frac{1}{1-x}$?

Jaki współczynnik stoi przy x^7 ?

Równy mocy zbioru $\{x^0x^0x^7, x^0x^2x^5, x^0x^4x^3, x^0, x^6, x^1, x^5x^0x^2, x^5x^2x^0\}$.

Równy mocy zbioru $\{(i, j, k) : 5i + 2j + k = 7\}$.

Liczba przedstawień n za pomocą dowolnej liczby składników

Niech $\langle r_n \rangle$ = liczba rozkładów n na składniki naturalne, gdy kolejność nie jest ważna

Liczba rozkładów n

Funkcja tworząca dla r_n to $\prod_{i=1}^{\infty} \frac{1}{1-x^i}$.

Liczba przedstawień n za pomocą różnych składników

Niech $\langle rr_n \rangle =$ liczba rozkładów n na różne składniki naturalne, gdy kolejność nie jest ważna.

Np. rozkład $5 = 1 + 2 + 2$ nie jest wliczany do rr_5 bo 2 występuje dwukrotnie.

Liczba rozkładów n na różne składniki

Funkcja tworząca dla rr_n to $\prod_{i=1}^{\infty} (1 + x^i)$.

Matematyka dyskretna (L)

Katarzyna Paluch

Instytut Informatyki, Uniwersytet Wrocławski

2021

Funkcje tworzące

Niech $\langle a_n \rangle$ będzie pewnym ciągiem a

$A(x) = \sum_{i=0}^{\infty} a_i x^i = a_0 + a_1 x + a_2 x^2 + \dots + a_i x^i + \dots$ jego funkcją tworzącą.

Funkcja tworzące dla ciągu:

$(a_0, 0, a_1, 0, a_2, 0, a_3, 0, \dots)$ to

$$\sum_{i=0}^{\infty} a_i x^i = a_0 + a_1 x^2 + a_2 x^4 + \dots + a_i x^{2i} + \dots = A(x^2).$$

Funkcja tworzące dla ciągu:

$(a_0, 0, 0, a_1, 0, 0, a_2, 0, 0, a_3, 0, 0, \dots)$ to

$$\sum_{i=0}^{\infty} a_i x^i = a_0 + a_1 x^3 + a_2 x^6 + \dots + a_i x^{3i} + \dots = A(x^3).$$

Funkcje tworzące

Niech $\langle a_n \rangle$ będzie pewnym ciągiem a

$A(x) = \sum_{i=0}^{\infty} a_i x^i = a_0 + a_1 x + a_2 x^2 + \dots + a_i x^i + \dots$ jego funkcją tworzącą.

Jak znaleźć funkcję tworzącą dla ciągu:

$(a_0, 0, a_2, 0, a_4, \dots)$?

Funkcje tworzące

Niech $\langle a_n \rangle$ będzie pewnym ciągiem a

$A(x) = \sum_{i=0}^{\infty} a_i x^i = a_0 + a_1 x + a_2 x^2 + \dots + a_i x^i + \dots$ jego funkcją tworzącą.

$A(-x) = \sum_{i=0}^{\infty} a_i (-x)^i = a_0 - a_1 x + a_2 x^2 - a_3 x^3 \dots + a_i (-x)^i + \dots$

Jak znaleźć funkcję tworzącą dla ciągu:
 $(a_0, 0, a_2, 0, a_4, \dots)$?

Funkcje tworzące

Niech $\langle a_n \rangle$ będzie pewnym ciągiem a

$A(x) = \sum_{i=0}^{\infty} a_i x^i = a_0 + a_1 x + a_2 x^2 + \dots + a_i x^i + \dots$ jego funkcją tworzącą.

$A(-x) = \sum_{i=0}^{\infty} a_i (-x)^i = a_0 - a_1 x + a_2 x^2 - a_3 x^3 \dots + a_i (-x)^i + \dots$

Jak znaleźć funkcję tworzącą dla ciągu:

$(a_0, 0, a_2, 0, a_4, \dots)$?

$$\frac{A(x) + A(-x)}{2}$$

Funkcje tworzące

Niech $\langle a_n \rangle$ będzie pewnym ciągiem a

$A(x) = \sum_{i=0}^{\infty} a_i x^i = a_0 + a_1 x + a_2 x^2 + \dots + a_i x^i + \dots$ jego funkcją tworzącą.

$A(-x) = \sum_{i=0}^{\infty} a_i (-x)^i = a_0 - a_1 x + a_2 x^2 - a_3 x^3 \dots + a_i (-x)^i + \dots$

Jak znaleźć funkcję tworzącą dla ciągu:
 $(0, a_1, 0, a_3, 0, \dots)$?

Funkcje tworzące

Niech $\langle a_n \rangle$ będzie pewnym ciągiem a

$A(x) = \sum_{i=0}^{\infty} a_i x^i = a_0 + a_1 x + a_2 x^2 + \dots + a_i x^i + \dots$ jego funkcją tworzącą.

$A(-x) = \sum_{i=0}^{\infty} a_i (-x)^i = a_0 - a_1 x + a_2 x^2 - a_3 x^3 \dots + a_i (-x)^i + \dots$

Jak znaleźć funkcję tworzącą dla ciągu:

$(0, a_1, 0, a_3, 0, \dots)$?

$$\frac{A(x) - A(-x)}{2}$$

Funkcje tworzące

Niech $\langle a_n \rangle$ będzie pewnym ciągiem a

$A(x) = \sum_{i=0}^{\infty} a_i x^i = a_0 + a_1 x + a_2 x^2 + \dots + a_i x^i + \dots$ jego funkcją tworzącą.

Jak znaleźć funkcję tworzącą dla ciągu:

$(0, a_1, 2a_2, 3a_3, 4a_4, \dots, ia_i, \dots)$?

Funkcje tworzące

Niech $\langle a_n \rangle$ będzie pewnym ciągiem a

$A(x) = \sum_{i=0}^{\infty} a_i x^i = a_0 + a_1 x + a_2 x^2 + \dots + a_i x^i + \dots$ jego funkcją tworzącą.

$$A'(x) = \sum_{i=0}^{\infty} i a_i x^{i-1} = a_1 + a_2 x + \dots + i a_i x^{i-1} + \dots$$

Jak znaleźć funkcję tworzącą dla ciągu:
 $(0, a_1, 2a_2, 3a_3, 4a_4, \dots, ia_i, \dots)$?

Funkcje tworzące

Niech $\langle a_n \rangle$ będzie pewnym ciągiem a

$A(x) = \sum_{i=0}^{\infty} a_i x^i = a_0 + a_1 x + a_2 x^2 + \dots + a_i x^i + \dots$ jego funkcją tworzącą.

$$A'(x) = \sum_{i=0}^{\infty} i a_i x^{i-1} = a_1 + a_2 x + \dots + i a_i x^{i-1} + \dots$$

Jak znaleźć funkcję tworzącą dla ciągu:

$(0, a_1, 2a_2, 3a_3, 4a_4, \dots, ia_i, \dots)$?

$$A'(x)x$$

Funkcje tworzące

Niech $\langle a_n \rangle$ będzie pewnym ciągiem a

$A(x) = \sum_{i=0}^{\infty} a_i x^i = a_0 + a_1 x + a_2 x^2 + \dots + a_i x^i + \dots$ jego funkcją tworzącą.

Jak znaleźć funkcję tworzącą dla ciągu:

$(0, a_1/1, a_2/2, a_3/3, a_4/4, \dots, a_i/i, \dots)$?

Funkcje tworzące

Niech $\langle a_n \rangle$ będzie pewnym ciągiem a

$A(x) = \sum_{i=0}^{\infty} a_i x^i = a_0 + a_1 x + a_2 x^2 + \dots + a_i x^i + \dots$ jego funkcją tworzącą.

$$\int A(x) dx = \sum_{i=0}^{\infty} \frac{a_i}{i+1} x^{i+1} + C$$

Jak znaleźć funkcję tworzącą dla ciągu:
 $(0, a_1/1, a_2/2, a_3/3, a_4/4, \dots, a_i/i, \dots)$?

Funkcje tworzące

Niech $\langle a_n \rangle$ będzie pewnym ciągiem o funkcji tworzącej

$$A(x) = \sum_{i=0}^{\infty} a_i x^i = a_0 + a_1 x + a_2 x^2 + \dots + a_i x^i + \dots$$

$$\int A(x) dx = \sum_{i=0}^{\infty} \frac{a_i}{i+1} x^{i+1} + C$$

Jak znaleźć funkcję tworzącą dla ciągu:
 $(0, a_1/1, a_2/2, a_3/3, a_4/4, \dots, a_i/i, \dots)$?

$$\int_0^1 \frac{A(x) - a_0}{x} dx$$

Graf nieskierowany to para zbiorów (V, E) , gdzie $E = \{\{u, v\} : u, v \in V\}$.
 V nazywamy zbiorem wierzchołków, a E krawędzi.

Pętla to krawędź postaci $\{v, v\}$.

Krawędzie równoległe - dwie lub więcej krawędzi łączące dwa wierzchołki u, v ($u \neq v$).

Graf $G = (V, E)$ jest **prosty** jeśli nie zawiera pętli ani krawędzi równoległych.

Grafy

Zastosowania grafów:

- znalezienie najkrótszej drogi,
- obliczenie przydziału zadań pracownikom,
- pokolorowanie mapy.

Grafy

Graf skierowany to para zbiorów (V, E) , gdzie $E = \{(u, v) : u, v \in V\}$.
 V nazywamy zbiorem wierzchołków, a E krawędzi skierowanych lub łuków.

Pętla to krawędź postaci $\{v, v\}$.

Krawedzie równoległe - dwie lub więcej krawędzi z u do v .

Graf $G = (V, E)$ jest prosty jeśli nie zawiera pętli ani krawędzi równoległych.

Stopień wierzchołka

Krawędź e jest **incydentna** do wierzchołka u , jeśli jeden z końców e to u .

Stopień wierzchołka u , oznaczany $\deg(u)$, to liczba krawędzi incydentnych do u .

(Każda pętla incydentna do u dokłada się do stopnia u liczbą 2.)

Lemat o uściskach dloni

Lemat

Niech $G = (V, E)$ będzie nieskierowanym grafem. Wtedy

$$\sum_{v \in V} \deg(v) = 2|E|.$$

Różne reprezentacje grafów

- listowa,
- za pomocą macierzy sąsiedztwa,
- za pomocą macierzy incydencji.

Izomorfizm grafów

Dwa grafy nieskierowane proste $G = (V, E)$ i $H = (V', E')$ są *izomorficzne* wtw, gdy \exists bijekcja $f : V \rightarrow V'$ taka, że

$$\forall_{u,v \in V} \{u, v\} \in E \Leftrightarrow \{f(u), f(v)\} \in E'.$$

Marszruta, ścieżka, droga

Marszrutą o długości k jest ciąg $(v_0, v_1, v_2, \dots, v_k)$ taki, że
 $\forall_{0 \leq i < k} \{v_i, v_{i+1}\} \in E$.

Droga to marszruta, w której żadna krawędź nie występuje dwukrotnie.

Ścieżka to marszruta, w której żaden wierzchołek nie występuje dwukrotnie.

Matematyka dyskretna (L)

Katarzyna Paluch

Instytut Informatyki, Uniwersytet Wrocławski

2020

Marszruta, ścieżka, droga

Marszrutą o długości k jest ciąg $(v_0, v_1, v_2, \dots, v_k)$ taki, że
 $\forall_{0 \leq i < k} \{v_i, v_{i+1}\} \in E$.

Droga to marszruta, w której żadna krawędź nie występuje dwukrotnie,
tzn. $\forall 1 \leq i < j \leq k - 1$ krawędzie $\{v_i, v_{i+1}\}, \{v_j, v_{j+1}\}$ są różne.

Ścieżka to marszruta, w której żaden wierzchołek nie występuje dwukrotnie.

Cykl to marszruta, w której pierwszy wierzchołek jest taki sam jak ostatni,
a poza tym, żaden wierzchołek nie występuje dwukrotnie.

Marszruta, ścieżka, droga

Marszrutą o długości k jest ciąg $(v_0, v_1, v_2, \dots, v_k)$ taki, że
 $\forall_{0 \leq i < k} \{v_i, v_{i+1}\} \in E$.

$u - v$ -marszruta to marszruta taka, że $v_0 = u$ i $v_k = v$.

Analogicznie definiujemy **$u - v$ -drogę** i **$u - v$ -ścieżkę**.

Marszruta/droga jest **zamknięta**, jeśli $v_0 = v_k$.

Zamknięta ścieżka to **cykl**.

Graf spójny

Nieskierowany graf $G = (V, E)$ jest **spójny**, jeśli "z każdego wierzchołka da się dojść do każdego innego", tzn. $\forall_{u,v \in V}$ w G istnieje $u - v$ -ścieżka (ścieżka łącząca u i v).

Podgrafem grafu $G = (V, E)$ jest dowolny graf $H = (V', E')$ taki, że $V' \subseteq V$ i $E' \subseteq E$.

Podgraf H jest *właściwy*, jeśli $G \neq H$.

Spójna składowa grafu G to dowolny podgraf spójny $H = (V', E')$ grafu G , który jest maksymalny ze względu na zawieranie tzn. taki, że nie istnieje podgraf spójny H' , którego podgrafem właściwym jest H .

Las i drzewo

Graf $G = (V, E)$ jest **acykliczny**, jeśli nie zawiera żadnego cyklu.

Las to acykliczny graf.

Drzewo to acykliczny graf spójny.

Spójne składowe lasu to drzewa.

Las składa się z drzew.

Pokażemy:

Drzewo to najmniejszy graf spójny: jeśli chcemy zbudować graf spójny G na zbiorze wierzchołków V , to G musi być drzewem.

Las, drzewa i liście

Graf $G = (V, E)$ jest **acykliczny**, jeśli nie zawiera żadnego cyklu.

Las to acykliczny graf.

Drzewo to acykliczny graf spójny.

Liść to wierzchołek o stopniu 1.

Czy każdy liść znajduje się na drzewie?

Czy drzewo składa się z liści?

Ile liści ma drzewo?

Las, drzewa i liście

Drzewo to acykliczny graf spójny.

Liść to wierzchołek o stopniu 1.

Dowolne drzewo o $n \geq 2$ wierzchołkach ma przynajmniej dwa liście.

Czy urywając z drzewa liść otrzymujemy drzewo?

Charakteryzacja drzewa

Niech $G = (V, E)$ będzie n -wierzchołkowym grafem nieskierowanym ($n \geq 1$). Wówczas następujące stwierdzenia są równoważne:

- ① G jest spójny i acykliczny (G jest drzewem),
- ② G jest spójny i ma $n - 1$ krawędzi.
- ③ G jest acykliczny i ma $n - 1$ krawędzi.
- ④ $\forall_{u,v \in V} \quad G$ zawiera dokładnie jedną $u - v$ -ścieżkę.

Most i cykl

Most to krawędź, której usunięcie zwiększa liczbę spójnych składowych grafu.

Żaden most nie leży na cyklu.

Grafy dwudzielne

Graf $G = (V, E)$ jest **dwudzielny** wtw, gdy istnieje podział zbioru wierzchołków V na zbiory A i B taki, że $\forall_{e \in E}$ jeden koniec e należy do A , a drugi do B .

Podział wierzchołków nie zawsze jest jednoznaczny.

Grafy dwudzielne

Graf $G = (V, E)$ jest **dwudzielny** wtw, gdy istnieje podział zbioru wierzchołków V na zbiory A i B taki, że $\forall_{e \in E}$ jeden koniec e należy do A , a drugi do B .

Czy dwudzielny graf G może zawierać cykl o nieparzystej długości?

Grafy dwudzielne - charakteryzacja

Graf dwudzielny

Graf $G = (V, E)$ jest dwudzielny wtw, gdy nie zawiera cyklu o nieparzystej długości.

Lemat

Każda zamknięta marszruta o nieparzystej długości zawiera cykl o nieparzystej długości.

Grafy o minimalnym stopniu k

Grafy o minimalnym stopniu k

Niech G będzie grafem prostym, w którym każdy wierzchołek ma stopień przynajmniej k . Wówczas G zawiera scieżkę o długości k . Jeśli $k \geq 2$, to G zawiera cykl o długości przynajmniej $k + 1$.

Matematyka dyskretna (L)

Katarzyna Paluch

Instytut Informatyki, Uniwersytet Wrocławski

2021

Drzewo rozpinające

Niech $G = (V, E)$ będzie grafem spójnym. Drzewo rozpinające grafu G to podgraf $T = (V, E')$, który jest drzewem. T zawiera wszystkie wierzchołki G .

Niech $G = (V, E)$ będzie grafem niekoniecznie spójnym. Las rozpinający grafu G to podgraf $F = (V, E')$, który jest lasem o liczbie spójnych składowych równej liczbie spójnych składowych G .

Niech $G = (V, E)$ będzie grafem spójnym o nieujemnych wagach na krawędziach $c : E \rightarrow R \geq 0$.

Drzewo rozpinające grafu G to podgraf $T = (V, E')$, który jest drzewem.
Waga drzewa rozpinającego $c(T) = \sum_{e \in E'} c(e)$.

Minimalne drzewo rozpinające (MST) grafu G to drzewo rozpinające G o minimalnej wadze.

Znajdowanie MST

Niech $G = (V, E)$ będzie grafem spójnym o nieujemnych wagach na krawędziach $c : E \rightarrow R \geq 0$.

Algorytm Kruskala:

$c(e_1) \leq c(e_2) \leq \dots \leq c(e_m)$ (sortujemy krawędzie względem wagi)
 $T \leftarrow \emptyset$

kolejno dla każdego $i, 1 \leq i \leq m$ wykonaj następujące:

jeśli dodanie e_i do T nie tworzy cyklu w T , dodaj e_i do T
(w p.p. nie dodawaj e_i do T)

Znajdowanie MST

Niech $G = (V, E)$ będzie grafem spójnym o nieujemnych wagach na krawędziach $c : E \rightarrow R \geq 0$.

Algorytm Kruskala:

$c(e_1) \leq c(e_2) \leq \dots \leq c(e_m)$ (sortujemy krawędzie względem wagi)
 $T \leftarrow \emptyset$

kolejno dla każdego $i, 1 \leq i \leq m$ wykonaj następujące:

jeśli dodanie e_i do T nie tworzy cyklu w T , dodaj e_i do T
(w p.p. nie dodawaj e_i do T)

Zarys implementacji.

Algorytm Prima

Niech $G = (V, E)$ będzie grafem spójnym o nieujemnych wagach na krawędziach $c : E \rightarrow R \geq 0$.

Algorytm Prima:

$T \leftarrow$ dowolny wierzchołek $u \in V$

Dopóki T nie jest drzewem rozpinającym wykonaj następujące:
spośród krawędzi o jednym wierzchołku w T a drugim poza
wybierz tę o najmniejszej wadze i dodaj ją do T

Algorytm Boruvki

Niech $G = (V, E)$ będzie grafem spójnym o nieujemnych wagach na krawędziach $c : E \rightarrow R \geq 0$.

Algorytm Boruvki:

$T \leftarrow V$ (wszystkie wierzchołki z V , zero krawędzi)

Dopóki T nie jest drzewem rozpinającym wykonaj następujące:

dla każdej spojnej składowej C_i grafu T wykonaj następujące:
spośród krawędzi o jednym wierzchołku w C_i a drugim poza
wybierz tę o najmniejszej wadze i oznacz ją jako $e(C_i)$
dodaj wszystkie krawędzie $e(C_i)$ do T

Skojarzenie (matching)

Niech $G = (V, E)$ będzie grafem spójnym. **Skojarzenie** grafu G to dowolny podzbiór krawędzi $M \subseteq E$ taki, że żadne dwie krawędzie z M nie mają wspólnego końca.

Zastosowania:

- rozlokowanie osób w pokojach 2-osobowych,
- przydział zadań pracownikom,
- przydział zadań maszynom.

Skajarzenie (matching)

Niech $G = (V, E)$ będzie grafem. Skajarzenie grafu G to dowolny podzbiór krawędzi $M \subseteq E$ taki, że żadne dwie krawędzie z M nie mają wspólnego końca.

Skajarzenie największe grafu G to skajarzenie o maksymalnej liczbie krawędzi.

Ścieżka alternująca

Niech $G = (V, E)$ będzie grafem spójnym, a M jakimś skojarzeniem w G . Wierzchołek $v \in V$ jest **skojarzony** w M , jeśli jest końcem jakieś krawędzi z M .

Wierzchołek $v \in V$ jest **nieskojarzony/ wolny** w M , jeśli żadna krawędź z M nie jest z nim incydentna.

Ścieżka P w grafie G jest **alternującą (względem M)** jeśli krawędzie na P na przemian należą i nie należą do M .

Ścieżka P w grafie G jest **powiększającą (względem M)**, jeśli jest alternującą (wzgl. M) i jej końce są nieskojarzone (w M).

Ścieżka alternująca

Niech $G = (V, E)$ będzie grafem spójnym, a M jakimś skojarzeniem w G . Wierzchołek $v \in V$ jest **skojarzony** w M , jeśli jest końcem jakieś krawędzi z M .

Wierzchołek $v \in V$ jest **nieskojarzony/ wolny** w M , jeśli żadna krawędź z M nie jest z nim incydentna.

Ścieżka P w grafie G jest **alternującą (względem M)** jeśli krawędzie na P na przemian należą i nie należą do M .

Ścieżka P w grafie G jest **powiększającą (względem M)**, jeśli jest alternującą (wzgl. M) i jej końce są nieskojarzone (w M).

Skojarzenie doskonałe/ pełne grafu G to skojarzenie, w którym każdy wierzchołek z V jest skojarzony.

Cykl alternujący

Cykl C w grafie G jest **alternującym (względem M)** jeśli krawędzie na C na przemian należą i nie należą do M .

Jaką długość ma cykl alternujący?

Twierdzenie Berge'a

Skojarzenie największe

Skojarzenie M grafu G jest największe wtw, gdy G nie zawiera ścieżki powiększającej względem M .

Warunek Halla

Niech $G = (V, E)$ będzie grafem a $W \subseteq V$ podzbiorem wierzchołków.
Sąsiedztwo W oznaczane jako $N(W)$ definujemy jako zbiór
 $\{v \in V : \exists_{w \in W} \{v, w\} \in E\}$.

Niech $G = (A \cup B, E)$ będzie grafem dwudzielnym.

Warunek Halla

Dla każdego $A' \subseteq A$ zachodzi $|N(A')| \geq |A'|$ oraz dla każdego $B' \subseteq B$ zachodzi $|N(B')| \geq |B'|$.

Skojarzenie doskonałe w grafie dwudzielnym

Niech $G = (A \cup B, E)$ będzie grafem dwudzielnym.

Warunek Halla

Dla każdego $A' \subseteq A$ zachodzi $|N(A')| \geq |A'|$ oraz dla każdego $B' \subseteq B$ zachodzi $|N(B')| \geq |B'|$.

Skojarzenie doskonałe w grafie dwudzielnym

Graf dwudzienny G zawiera skojarzenie doskonałe wtw, gdy spełniony jest w nim warunek Halla.

Skojarzenie doskonałe w grafie dwudzielnym

W pewnej grupie muzykujących osób Ania gra na skrzypcach, harfie, kontrabasie i wiolonczeli, Bartek gra na harfie i fortepianie, Cezary gra na fortepianie, Dąbrówka gra na harfie i Elwira gra na kontrabasie, skrzypcach, wiolonczeli i harfie.
Chcieliby zagrać utwór na fortepian, skrzypce, wiolonczelę, kontrabas i harfę. Czy uda im się dobrać skład?

Matematyka dyskretna (L)

Katarzyna Paluch

Instytut Informatyki, Uniwersytet Wrocławski

2021

Królewiec XVIII wiek

Święty Mikołaj jest w Królewcu w XVIII wieku, gdzie układ mostów jest taki jak na whiteboardzie. Chciałby zaplanować przejażdżkę saniami, w której na każdym moście znalazłby się dokładnie raz. Punkt startu i zakończenia jest taki sam, ale może go sobie wybrać.

Czy jego zamierzenie jest wykonalne?

Królewiec XVIII wiek

Święty Mikołaj jest w Królewcu w XVIII wieku, gdzie układ mostów jest taki jak na whiteboardzie. Chciałby zaplanować przejażdżkę saniami, w której na każdym moście znalazłby się dokładnie raz. Punkt startu i zakończenia jest taki sam, ale może go sobie wybrać.

Czy jego zamierzenie jest wykonalne?

A jeśli punkty startu i końcowy nie muszą być takie same?

Święty Mikołaj w mieście M

Święty Mikołaj jest w mieście M , którego rozkład ulic da się przedstawić za pomocą grafu spójnego $G = (V, E)$, nieskierowanego, gdyż każda ulica jest dwukierunkowa. Chciałby rozwieźć prezenty mieszkańcom wszystkich ulic (przy każdej ktoś mieszka) tak, by każdą ulicą przejechać dokładnie raz. Od czego zależy to, czy mu się to uda? Punkt startu i zakończenia może być taki sam, może być też różny.

Droga i cykl Eulera

Niech $G = (V, E)$ będzie grafem spójnym nieskierowanym, niekoniecznie prostym.

Droga Eulera grafu G to droga (krawędzie się nie powtarzają, wierzchołki mogą), która zawiera każdą krawędź $e \in E$.

Cykl Eulera grafu G to droga zamknięta (wierzchołek startowy jest taki sam jak końcowy), która zawiera każdą krawędź $e \in E$.

Warunki istnienia drogi/cyklu Eulera

Spójny graf G posiada drogę Eulera wtw, gdy zawiera 0 lub 2 wierzchołki o stopniu nieparzystym. Spójny graf G posiada cykl Eulera wtw, gdy wszystkie jego wierzchołki mają stopień parzysty.

Święty Mikołaj w Finlandii

Święty Mikołaj jest w Finlandii i rozkład osad, do których chciałby przyjechać da się przedstawić za pomocą grafu spójnego $G = (V, E)$ nieskierowanego. Chciałby rozwieźć prezenty mieszkańcom wszystkich osad odwiedzając każdą osadę dokładnie raz. Od czego zależy to, czy mu się to uda? Punkt startu i zakończenia może być taki sam, może być też różny.

Ścieżka i cykl Hamiltona

Niech $G = (V, E)$ będzie grafem spójnym nieskierowanym.

Ścieżka Hamiltona grafu G to ścieżka (wierzchołki się nie powtarzają), która zawiera każdy wierzchołek $v \in V$.

Cykl Hamiltona grafu G to cykl (wierzchołki się nie powtarzają), który zawiera każdy wierzchołek $v \in V$.

Ścieżka/cykl Hamiltona

Sprawdzenie, czy graf $G = (V, E)$ zawiera ścieżkę lub cykl Hamiltona jest problemem trudnym obliczeniowo - jest to problem NP -trudny.

Święty Mikołaj w Finlandii

Święty Mikołaj jest w Finlandii i rozkład osad, do których chciałby przyjechać da się przedstawić za pomocą nieskierowanego grafu spójnego. (Każde dwie osady są bezpośrednio połączone.) Odległości między osadami są zapisane jako funkcja wagi $c : E \rightarrow R \geq 0$. Chciałby obejrzeć wszystkie osady pokonując w sumie jak najmniejszą odległość.

Dodatkowo:

- ① punkt startu = punkt końcowy,
- ② skrót zawsze się opłaca - $\forall_{u,v,w \in V} c(u, v) \leq c(u, w) + c(w, v)$.

Innymi słowy, Św. Mikołaj chciałby obliczyć **najmniejszy wagowo cykl Hamiltona**.

Święty Mikołaj w Finlandii

Święty Mikołaj jest w Finlandii i rozkład osad, do których chciałby przyjechać da się przedstawić za pomocą nieskierowanego grafu spójnego. (Każde dwie osady są bezpośrednio połączone.) Odległości między osadami są zapisane jako funkcja wagi $c : E \rightarrow R \geq 0$. Chciałby obejrzeć wszystkie osady pokonując w sumie jak najmniejszą odległość.

Dodatkowo:

- ① punkt startu = punkt końcowy,
- ② skrót zawsze się opłaca - $\forall_{u,v,w \in V} c(u, v) \leq c(u, w) + c(w, v)$.

Niech OPT oznacza sumaryczną długość optymalnej trasy Św. Mikołaja. Jak się ma $MST(G)$ do OPT ?

Święty Mikołaj w Finlandii

Święty Mikołaj jest w Finlandii i rozkład osad, do których chciałby przyjechać da się przedstawić za pomocą nieskierowanego grafu spójnego. (Każde dwie osady są bezpośrednio połączone.) Odległości między osadami są zapisane jako funkcja wagi $c : E \rightarrow R \geq 0$. Chciałby obejrzeć wszystkie osady pokonując w sumie jak najmniejszą odległość.

Dodatkowo:

- ① punkt startu = punkt końcowy,
- ② skrót zawsze się opłaca - $\forall_{u,v,w \in V} c(u, v) \leq c(u, w) + c(w, v)$.

Niech OPT oznacza sumaryczną długość optymalnej trasy Św. Mikołaja. Jak się ma waga $MST(G)$ do OPT ?

$$c(MST(G)) \leq OPT$$

Święty Mikołaj w Finlandii - trasa

Algorytm znajdujący trasę Św. Mikołaja:

- ① Oblicz $MST(G)$.
- ② Podwój $MST(G)$ otrzymując T^2 .
- ③ Znajdź cykl Eulera C_E podwojonego $MST(G)$, czyli T^2 .
- ④ Skróć C_E do cyklu Hamiltona.

Święty Mikołaj w Finlandii - trasa

Algorytm znajdujący trasę Św. Mikołaja:

- ① Oblicz $MST(G)$.
- ② Podwój $MST(G)$ otrzymując.
- ③ Znajdź cykl Eulera C_E podwojonego $MST(G)$.
- ④ Skróć C_E do cyklu Hamiltona.

Obliczona trasa ma wagę $\leq 2OPT$.

Święty Mikołaj w Finlandii - trasa

Algorytm Christofidesa znajdujący trasę Św. Mikołaja:

- ① Oblicz $MST(G)$.
- ② Oblicz najmniejsze wagowo skojarzenie pełne M na podgrafie zawierającym wierzchołki V^- , które mają st. nieparzysty w $MST(G)$.
- ③ Znajdź cykl Eulera C_E multigrafu $MST(G) + M$.
- ④ Skróć C_E do cyklu Hamiltona.

Obliczona trasa ma wagę $\leq \frac{3}{2}OPT$.

Matematyka dyskretna (L)

Katarzyna Paluch

Instytut Informatyki, Uniwersytet Wrocławski

2022

Najkrótsze ścieżki

Niech $G = (V, E)$ będzie grafem spójnym.

Jak znaleźć najkrótszą ścieżkę z s do t ?

Jak znaleźć najkrótszą ścieżkę z s do v dla każdego wierzchołka $v \in V$?

Najkrótsze ścieżki

Niech $G = (V, E)$ będzie grafem spójnym o nieujemnych wagach na krawędziach $c : E \rightarrow R \geq 0$.

Waga ścieżki P to suma wag krawędzi leżących na P .

Najlżejsza / najkrótsza (względem c) ścieżka z s do t to ta ze ścieżek z s do t , która ma najmniejszą wagę.

Jak znaleźć najkrótszą (wzgl. c) ścieżkę z s do t ?

Jak znaleźć najkrótszą (wzgl. c) ścieżkę z s do v dla każdego wierzchołka $v \in V$?

Najkrótsze ścieżki

Niech $G = (V, E)$ będzie grafem spójnym o nieujemnych wagach na krawędziach $c : E \rightarrow R \geq 0$,
a s ustalonym wierzchołkiem z V .

Niech $S \subseteq V$. Ścieżka P z s do v jest **prawie S -owa / osiągalna bezpośrednio z S** jeśli wszystkie wierzchołki na P oprócz v są w S .
 $d(v)$ - waga najkrótszej ścieżki z s do v
 $t(v)$ - waga najkrótszej prawie S -owej ścieżki z s do v ; jeśli takiej ścieżki nie ma, to $t(v) = \infty$

Algorytm Dijkstry

$G = (V, E)$ - graf spójny; $c : E \rightarrow R \geq 0$, $s \in V$

$d(v)$ - waga najkrótszej ścieżki z s do v

$t(v)$ - waga najkrótszej prawie S -owej ścieżki z s do v ; jeśli takiej ścieżki nie ma, to $t(v) = \infty$

$S \leftarrow \{s\}$, $d(s) \leftarrow 0$

dla każdego sąsiada v wierzchołka s : $t(v) \leftarrow c(s, v)$

dla pozostałych wierzchołków: $t(v) \leftarrow \infty$

dopóki $S \neq V$ wykonaj:

$u \leftarrow \operatorname{argmin}\{t(u) : u \notin S\}$

dodaj u do S

zaktualizuj wartości $t(v)$:

dla każdego sąsiada $v \notin S$ wierzchołka u :

$t(v) \leftarrow \min\{t(v), d(u) + c(u, v)\}$

Algorytm Dijkstry

Jak zmodyfikować algorytm Dijkstry, by znajdować najkrótsze ścieżki a nie tylko wagi najkrótszych ścieżek?

Czy algorytm ten działa również:

- w grafach skierowanych?
- gdy wagi krawędzi mogą być ujemne?

Warunek Halla

Niech $G = (V, E)$ będzie grafem a $W \subseteq V$ podzbiorem wierzchołków.
Sąsiedztwo W oznaczane jako $N(W)$ definujemy jako zbiór
 $\{v \in V : \exists_{w \in W} \{v, w\} \in E\}$.

Niech $G = (A \cup B, E)$ będzie grafem dwudzielnym.

Warunek Halla

Dla każdego $A' \subseteq A$ zachodzi $|N(A')| \geq |A'|$ oraz dla każdego $B' \subseteq B$ zachodzi $|N(B')| \geq |B'|$.

Skojarzenie doskonałe w grafie dwudzielnym

Niech $G = (A \cup B, E)$ będzie grafem dwudzielnym.

Warunek Halla

Dla każdego $A' \subseteq A$ zachodzi $|N(A')| \geq |A'|$ oraz dla każdego $B' \subseteq B$ zachodzi $|N(B')| \geq |B'|$.

Skojarzenie doskonałe w grafie dwudzielnym

Graf dwudzienny G zawiera skojarzenie doskonałe wtw, gdy spełniony jest w nim warunek Halla.

Pokrycie wierzchołkowe

Niech $G = (V, E)$ będzie grafem.

Pokrycie wierzchołkowe grafu G to dowolny podzbiór $V' \subseteq V$ taki, że każda krawędź z E ma przynajmniej jeden z końców w V' .

Najmniejsze pokrycie wierzchołkowe grafu G to spośród pokryć wierzchołkowych G , które zawiera najmniej wierzchołków.

Pokrycie wierzchołkowe a skojarzenie

Niech $G = (V, E)$ będzie grafem.

Pokrycie wierzchołkowe grafu G to dowolny podzbiór $V' \subseteq V$ taki, że każda krawędź z E ma przynajmniej jeden z końców w V' .

Niech M będzie jakimś skojarzeniem G a W jakimś pokryciem wierzchołkowym.

Czy możemy jakoś porównać $|M|$ i $|W|$? $|M| \leq |W|$? $|M| \geq |W|$

Pokrycie wierzchołkowe a skojarzenie

Niech $G = (V, E)$ będzie grafem.

Niech M będzie jakimś skojarzeniem G a W jakimś pokryciem wierzchołkowym.

Wtedy $|M| \leq |W|$.

Pokrycie wierzchołkowe a skojarzenie

Niech $G = (V, E)$ będzie grafem.

Niech M_{max} będzie największym skojarzeniem G a W_{min} najmniejszym pokryciem wierzchołkowym.

Wtedy $|M_{max}| \leq |W_{min}|$.

A może zachodzi równość?

Twierdzenie Koeniga

Niech $G = (V, E)$ będzie grafem dwudzielnym, M_{max} największym skojarzeniem G a W_{min} najmniejszym pokryciem wierzchołkowym. Wtedy $|M_{max}| = |W_{min}|$.

Matematyka dyskretna (L)

Katarzyna Paluch

Instytut Informatyki, Uniwersytet Wrocławski

2022

Ścieżka i cykl Hamiltona

Niech $G = (V, E)$ będzie grafem spójnym nieskierowanym.

Ścieżka Hamiltona grafu G to ścieżka, która zawiera każdy wierzchołek $v \in V$.

Cykl Hamiltona grafu G to cykl, który zawiera każdy wierzchołek $v \in V$.

Ścieżka/cykl Hamiltona

Sprawdzenie, czy graf $G = (V, E)$ zawiera ścieżkę lub cykl Hamiltona jest problemem trudnym obliczeniowo - jest to problem NP -trudny.

Warunki konieczne na istnienie cyklu Hamiltona

- Jeśli graf $G = (A \cup B, E)$ jest dwudzielny, to warunkiem koniecznym na istnienie cyklu Hamiltona jest: $|A| = |B|$.
- Jeśli graf $G = (V, E)$ zawiera cykl Hamiltona, to dla dowolnego zbioru $S \subseteq V$, graf $G - S$ (powstały po usunięciu wierzchołków z S wraz z incydentnymi krawędziami) zawiera co najwyżej $|S|$ spójnych składowych.

Twierdzenie Diraca

Jeśli $G = (V, E)$ jest grafem prostym o co najmniej trzech wierzchołkach i minimalnym stopniu wierzchołka $\delta(G) \geq |V|/2$, to G zawiera cykl Hamiltona.

Twierdzenie Ore'a

Jeśli $G = (V, E)$ jest grafem prostym o co najmniej trzech wierzchołkach i takim, że dla każdych dwóch wierzchołków u i v niepołączonych krawędzią zachodzi $\deg(u) + \deg(v) \geq |V|$, to G zawiera cykl Hamiltona.

Kolorowanie grafu

Niech $G = (V, E)$ będzie grafem prostym.

Kolorowaniem wierzchołkowym grafu G nazywamy funkcję

$f : V \rightarrow$ Kolory taką, że $\forall_{(u,v) \in E} f(u) \neq f(v)$.

$\chi(G)$ - liczba chromatyczna G to najmniejsza liczba kolorów, jaką można pokolorować graf G .

Ille wynosi $\chi(G)$, jeśli G jest:

- grafem dwudzielnym?
- kliką n - wierzchołkową?
- cyklem o długości $2n + 1$?

Liczba chromatyczna - własności

$\omega(G)$ to wielkość największej kliki zawartej w G .

Zauważamy, że $\chi(G) \geq \omega(G)$.

Czy istnieją grafy, dla których zachodzi: $\chi(G) > \omega(G)$?

Algorytm sekwencyjny kolorowania grafu

Niech $Kolory = \{1, 2, 3, \dots\}$.

$G = (V, E)$

Algorytm sekwencyjny:

- ① Ustaw wierzchołki z V w pewien ciąg.
- ② Dla każdego wierzchołka v w kolejności dyktowanej przez ciąg wykonaj:

przypisz wierzchołkowi v najmniejszą liczbę naturalną spośród takich, które nie są przypisane żadnemu sąsiadowi v .

Liczba chromatyczna - własności

$\Delta(G)$ to największy stopień wierzchołka w G .

$$\chi(G) \leq \Delta(G) + 1$$

Twierdzenie Brooksa

Twierdzenie Brooksa

Jeśli G nie jest kliką ani nieparzystym cyklem, to $\chi(G) \leq \Delta(G)$.

Kolorowanie grafów przedziałowych

Zbiór wierzchołków to odcinki na prostej. Dwa odcinki są połączone krawędzią, jeśli się przecinają.

Matematyka dyskretna (L)

Katarzyna Paluch

Instytut Informatyki, Uniwersytet Wrocławski

2022

3 osobników leśnych

Trzej zawzięci wrogowie X , Y , Z mieszkają w lesie. Każdy z nich chce poprowadzić ścieżkę do gazu, wody i elektryczności (każdy z tych trzech zasobów jest w jednym miejscu). Czy istnieje taki sposób poprowadzenia tych 9 sciezek, by żadne dwie się nie przecinały?

3 osobników leśnych

Trzej zawzięci wrogowie X , Y , Z mieszkają w lesie. Każdy z nich chce poprowadzić ścieżkę do gazu, wody i elektryczności (każdy z tych trzech zasobów jest w jednym miejscu). Czy istnieje taki sposób poprowadzenia tych 9 scieżek, by żadne dwie się nie przecinały?

Innymi słowy: czy da się narysować graf $K_{3,3}$ na płaszczyźnie tak, by żadne dwie krawędzie się nie przecinały?

Graf planarny

Graf G jest **planarny**, gdy da się go narysować na płaszczyźnie w taki sposób, by żadne dwie krawędzie się nie przecinały.

Co to znaczy narysować graf na płaszczyźnie?

Rysunek grafu

Łamana (linia wielokątną, linia łamana) to ciąg skończenie wielu odcinków, z których każdy zaczyna się tam, gdzie poprzedni kończy; poza tym żadne dwa odcinki nie mają punktów wspólnych.

Rysunek grafu $G = (V, E)$ na płaszczyźnie to funkcja różnowartościowa f :

- ① odwzorowująca każdy wierzchołek $v \in V$ na punkt $f(v)$ płaszczyzny oraz
- ② każdą krawędź (u, v) na łamanałą łączącą $f(u)$ z $f(v)$.

Mówimy, że rysunek **nie ma przecięć**, jeśli dla dowolnych dwóch krawędzi e, e' $f(e) \cap f(e')$ może zawierać jedynie obrazy wspólnych konców e i e' .

Graf G jest **planarny**, jeśli posiada rysunek bez przecięć.

Konkretny rysunek bez przecięć grafu G nazywamy **grafem płaskim**.

Graf płaski

Ściana w grafie płaskim G to spójny obszar płaszczyzny po usunięciu linii reprezentujących krawędzie. Innymi słowy, ściana to zbiór punktów płaszczyzny, które da się połączyć krzywą nieprzecinającą żadnej krawędzi.

Granica ściany zawiera krawędzie "styczne" z tą ścianą.

Długość granicy ściany to długość najkrótszej zamkniętej marszruty przechodzącej przez wszystkie krawędzie granicy tej ściany.

Niech f_i oznacza długość granicy i -tej ściany grafu planarnego $G = (V, E)$, a I liczbę ścian G . Wtedy:

$$\sum_{i=1}^I f_i = 2|E|.$$

Twierdzenie Jordana

Twierdzenie Jordana

Zamknięta nieprzecinająca się łamana C o skończonej liczbie odcinków dzieli płaszczyznę na dokładnie dwie ściany, z których każda ma C jako granicę.

Graf dualny

Niech $G = (V, E)$ będzie grafem planarnym. Graf dualny G^* dla grafu płaskiego G tworzy się następująco:

- Dla każdej ściany (włącznie z ścianą zewnętrzną) grafu G dodajemy wierzchołek.
- Jeśli dwie ściany mają wspólną krawędź e , łączymy wierzchołki utworzone w poprzednim kroku odpowiednie dla sąsiadujących ścian krawędzią przecinającą tylko krawędź e .

Graf dualny grafu planarnego G nie jest wyznaczony jednoznacznie - zależy od rysunku G .

Wzór Eulera

Niech G będzie spójnym grafem planarnym (niekoniecznie prostym) o n wierzchołkach, m krawędziach i f ścianach. Wówczas $n - m + f = 2$.

Krawędzie w grafie planarnym

Liczba krawędzi grafu planarnego

Niech G będzie prostym grafem planarnym o $n \geq 3$ wierzchołkach. Wówczas liczba krawędzi m tego grafu nie przekracza $3n - 6$. Jeśli dodatkowo, G nie zawiera żadnego trójkąta, to $m \leq 2n - 4$.

Grafy homeomorficzne

Grafy G i H są **homeomorficzne**, gdy jeden można przekształcić do drugiego za pomocą skończonej liczby operacji następujących dwóch typów:

- ① zamian krawędzi na ścieżkę o długości 2, tj. w ten sposób dodajemy również jeden nowy wierzchołek,
- ② zamiana ścieżki $P = (u, v, w)$ takiej, że v ma stopień 2 na krawędź (u, w) , jednocześnie usuwając v .

Twierdzenie Kuratowskiego

Twierdzenie Kuratowskiego [1930]

Graf G jest planarny wtedy i tylko wtedy, gdy nie zawiera podgrafu homeomorficznego z $K_{3,3}$ lub K_5 .

Kolorowanie grafu planarnego

Twierdzenie Heawooda [1890]

Każdy graf planarny jest 5-kolorowalny.

Matematyka dyskretna (L)

Katarzyna Paluch

Instytut Informatyki, Uniwersytet Wrocławski

2022

Przepływ w sieci

Sieć to graf skierowany (digraf) $D = (V, E)$ z dwoma wyróżnionymi wierzchołkami $s, t \in V$ zwanymi *źródłem* i *ujściem* i funkcją przepustowości $c : E \rightarrow R \geq 0$ na krawędziach.

Niech $f : E \rightarrow R$.

dla $v \in V$ definiujemy $f^+(v) = \sum_{e=(v,w): e \in E, w \in V} f(v, w)$ oraz
 $f^-(v) = \sum_{e=(w,v): e \in E, w \in V} f(w, v)$

Funkcja f jest **przepływem**, jeśli

- spełnia warunki przepustowości $\forall_{e \in E} 0 \leq f(e) \leq c(e)$, oraz
- jeśli spełnia warunek zachowania przepływu:
 $\forall_{v \in V \setminus \{s, t\}} f^+(v) = f^-(v).$

Wartość przepływu f , oznaczana jako $|f|$ to $f^-(t) - f^+(t)$.

Ścieżka powiększająca

Ścieżka powiększająca P dla przepływu f to ścieżka postaci $(s = v_0, e_1, v_1, e_2, v_2, \dots, e_k, v_k, e_{k+1}, t = v_{k+1})$ taka, że:

- $\forall 0 \leq i \leq k \ e_{i+1} \in E \wedge (e_{i+1} = (v_i, v_{i+1}) \vee e_{i+1} = (v_{i+1}, v_i))$
- $\forall 0 \leq i \leq k \ e_{i+1} = (v_i, v_{i+1}) \Rightarrow f(e_{i+1}) < c(e_{i+1})$ (krawędź w przód)
- $\forall 0 \leq i \leq k \ e_{i+1} = (v_{i+1}, v_i) \Rightarrow f(e_{i+1}) > 0$ (krawędź wsteczna).

Luz ścieżki powiększającej P to minimum z dwóch minimów:

$\min\{c(e) - f(e)\}$ po wszystkich krawędziach w przód ścieżki oraz
 $\min\{f(e)\}$ po wszystkich krawędziach wstecznych.

Zastosowanie ścieżki powiększającej

Niech $P = (s = v_0, e_1, v_1, e_2, v_2, \dots, e_k, v_k, e_{k+1}, t = v_{k+1})$ to ścieżka powiększająca dla przepływu f o luzie ϵ .

Zastosować P do przepływu f oznacza funkcję f' taką, że

- dla $e \in E \setminus P$: $f'(e) = f(e)$,
- dla $e \in P$ w przód: $f'(e) = f(e) + \epsilon$,
- dla $e \in P$ wstecznej: $f'(e) = f(e) - \epsilon$

Lemat

f' jest przepływem takim, że $|f'| = |f| + \epsilon$.

Algorytm Forda-Fulkersona

$D = (V, E)$ - digraf spójny; $c : E \rightarrow \mathbb{R} \geq 0$ $s, t \in V$

$\forall_{e \in E} f(e) \leftarrow 0$

dopóki istnieje ścieżka powiększająca P dla f **wykonaj:**
zastosuj P do f , otrzymując f'
 $f \leftarrow f'$

Znajdowanie ścieżki powiększającej:

$$R \leftarrow \{s\}$$

dopóki można **wykonaj**:

jeśli istnieje krawędź $e = (u, v) : u \in R, v \notin R, f(e) < c(e)$,
to dodaj v do R ,

jeśli istnieje krawędź $e = (v, u) : u \in R, v \notin R, f(e) > 0$,
to dodaj v do R ,

Jeśli R zawiera t , to znaczy, że istnieje ścieżka pow. P dla f .

Algorytm F-F

- Czy algorytm F-F zawsze się kończy?
- Czy znajduje maksymalny przepływ?

Przepustowość przekroju

[S, T] to **s – t przekrój**, jeśli $s \in S$, $t \in T$, $S \cup T = V$, $S \cap T = \emptyset$.
przepustowość przekroju $c([S, T]) = \sum_{e=(u,v) \in E: u \in S, v \in T} c(e)$.

Lemat

Niech $U \subset V$. Wtedy $f^+(U) - f^-(U) = \sum_{v \in U} f^+(v) - f^-(v)$.
Dla dowolnego s – t przekroju [S, T] zachodzi $|f| \leq c([S, T])$.

Maksymalny przepływ-minimalne cięcie

Twierdzenie

Przepływ f obliczony przez algorytm Forda-Fulkersona ma wartość równą przepustowości pewnego $s - t$ przekroju. Zatem jest maksymalny.

Przepływ całkowitoliczbowy

Jeśli przepustowość każdej krawędzi w sieci jest liczbą całkowitą, to istnieje przepływ f maksymalny, który jest całkowitoliczbowy.

Zastosowania przepływów

- znajdowanie największego skojarzenia w grafach dwudzielnych
- znajdowanie największego b -skojarzenia w grafach dwudzielnych

Niech $b : V \rightarrow N$. Wtedy $M \subseteq E$ jest **b -skojarzeniem**, jeśli
 $\forall_{v \in V} \deg_M(v) \leq b(v)$ (liczba krawędzi z M incydentnych do v nie przekracza $b(v)$).

Zastosowania przepływów

n studentów lat I – III należy do k różnych kół. Każdy student może należeć do dowolnej liczby kół. Rektor chciałby wyznaczyć reprezentację, w której każde koło reprezentowane jest przez jednego ze studentów oraz liczba studentów w reprezentacji z każdego roku jest taka sama i wynosi $\lceil \frac{k}{3} \rceil$. Skonstruuj algorytm, który taką reprezentację znajduje.

Zastosowania przepływów

Organizowany jest turniej n osób, w którym każdy gra z każdym. Każda rozgrywka kończy się wygraną dokładnie jednej z osób - nie ma remisów. Wynik turnieju to graf pełny skierowany na n wierzchołkach, w którym krawędź skierowana z u do v oznacza wygraną wierzchołka u w meczu z v . Czy możliwy jest wynik turnieju, w którym różnica liczby wygranych dowolnych dwóch osób jest niewiększa od 1?

Ogólniej, czy dla każdego ciągu n liczb całkowitych dodatnich a_1, a_2, \dots, a_n takiego, że $\sum_{i=1}^n a_i = \binom{n}{2}$ istnieje wynik turnieju taki, że osoba i wygrała dokładnie a_i pojedynków?

W obu przypadkach pokaż algorytm znajdowania takiego rozkładu, o ile istnieje.

$$\begin{aligned}
|\bigcup_{i=1}^{n+1} A_i| &= \left| (\bigcup_{i=1}^n A_i) \cup A_{n+1} \right| = \left| \bigcup_{i=1}^n A_i \right| + |A_{n+1}| - \left| \left(\bigcup_{i=1}^n A_i \right) \cap A_{n+1} \right| \\
&= \left| \bigcup_{i=1}^n A_i \right| + |A_{n+1}| - \left| \bigcup_{i=1}^n (A_i \cap A_n) \right| \quad (\text{Vorordnung } \cap \text{ vor } \cup)
\end{aligned}$$

$$B_i = A_i \cap A_n$$

$$= \left| \bigcup_{i=1}^n A_i \right| + |A_{n+1}| - \left| \bigcup_{i=1}^n B_i \right|$$

\geq induktiv

$$\sum_{k=1}^n (-1)^{k-1} \sum_{\substack{\emptyset \neq I \subseteq \{1, 2, \dots, n\} \\ |I|=k}} \left| \bigcap_{i \in I} A_i \right| + |A_{n+1}| - \sum_{k=1}^n (-1)^{k-1} \sum_{\substack{\emptyset \neq I \subseteq \{1, 2, \dots, n\} \\ |I|=k}} \left| \bigcap_{i \in I} B_i \right|$$

X

$\left| \bigcap_{i \in I} B_i \right|$

$= \left| \bigcap_{i \in I} (A_i \cap A_{n+1}) \right| = \left| \left(\bigcap_{i \in I} A_i \right) \cap A_{n+1} \right|$

Y

$$-Y = \sum_{k=2}^{n+1} (-1)^k \sum_{\substack{\emptyset \neq I \subseteq \{1, 2, \dots, n\} \\ |I|=k}} \left| \left(\bigcap_{i \in I} A_i \right) \cap A_{n+1} \right|$$

$$\begin{aligned}
&= \sum_{k=2}^{n+1} (-1)^k \sum_{\substack{\emptyset \neq I \subseteq \{1, 2, \dots, n+1\} \\ n+1 \in I \\ |I|=k+1}} \left| \bigcap_{i \in I} A_i \right|
\end{aligned}$$

$$|A_{n+1}| - Y = \sum_{k=1}^{n+1} (-1)^k \sum_{\substack{\emptyset \neq I \subseteq \{1, 2, \dots, n+1\} \\ i+1 \in I}} |\bigcap_{i \in I} A_i|$$

$|I| = k+1$

$$\sum_{k=1}^n (-1)^{k-1} \sum_{\substack{\emptyset \neq I \subseteq \{1, 2, \dots, n\} \\ |I| = k}} |\bigcap_{i \in I} A_i| + \sum_{k=1}^{n+1} (-1)^k \sum_{\substack{\emptyset \neq I \subseteq \{1, 2, \dots, n+1\} \\ n+1 \in I}} |\bigcap_{i \in I} A_i|$$

$|I| = k+1$

\curvearrowleft \times

\curvearrowleft \times'

wszystkie precise, w których występuje A_{n+1}

$$= \sum_{k=1}^{n+1} (-1)^{k-1} \sum_{\substack{\emptyset \neq I \subseteq \{1, 2, \dots, n+1\} \\ |I| = k}} |\bigcap_{i \in I} A_i|$$