Webinarium przeprowadzone w ramach projektu "Mistrzostwa w Algorytmice i Programowaniu - Uczniowie", **finansowanego przez:**

Piotr Chrząstowski-Wachtel
Uniwersytet Warszawski

Indukcja matematyczna

Indukcja – zupełnie nowy sposób rozumowania

- Indukcja matematyczna jest pomysłem starym jak świat, i na tyle prostym, że dość późno zauważono, że jest w ogóle o czym rozmawiać i że wcale nie jest oczywiste, jak wielką siłę daje.
- Pierwszy, który sformalizował pojęcie indukcji matematycznej był Blaise (Błażej) Pascal – użył jej do pokazania paru własności “swojego” trójkąta.

O co chodzi?

- Pomysł jest taki: zamiast mówić o pewnych własnościach obiektów wprost, zróbmy to “informatycznie” - powiedzmy coś o własności jakegoś wybranego obiektu, a następnie zlećmy każdemu obiektowi, który ma tę własność, przekonanie innych obiektów, że też tę własność mają.
- Jeśli uda się nam w ten sposób dotrzeć do wszystkich obiektów, to możemy powiedzieć, że powiedzieliśmy wszystko o wszystkim, co chcieliśmy.

Co to znaczy “dotrzeć”?

- Trzeba to ustalić. Normalnie robi się to tak, że ustala sie relację między elementami mówiącą, kto kogo ma “powiadamiać”.
- Najprościej robi się to w liczbach naturalnych: relacja powiadamiania może być relacją “+1” (czyli n powiadamia n+1 dla każdego n). Jeśli powiadomimy o czymś 0, a następnie powiemy “niech każdy powiadomiony powiadomi liczbę o 1 większą, to wiadomość do wszystkich dotrze.

Co to jest to “coś”, o czym powiadamiamy?

- Może to być definicja jakiegoś ciągu wartości. Ustalmy, że ciąg, to jest $a_0,a_1,a_2,...$, i że każda liczba naturalna jest odpowiedzialna za “swój” wyraz ciągu, zatem dla każdego n liczba n jest odpowiedzialna za określenie a_n.

Przykłady definicji indukcyjnych

- Dla pewnych wartości początkowych a,d:
  - $a_0=a$
  - $a_{n+1}=a_n+d$
- Zatem jeśli np.
  - a=3, d=5, to mamy ciąg 3, 8, 13, 18, 23 ...
  - a=-6, d=2, to mamy ciąg -6, -4, -2, 0, 2...
  - a=0, d=0, to mamy ciąg 0, 0, 0, ...
- Można też ogólnie podać wzór $a_n=a_0+nd$, (albo $a_n=a_1+(n-1)d$),co będzie stanowiło inną definicję.

Definicje indukcyjne - cd

- Dla pewnych wartości początkowych a,q:
  - $a_0=a$
  - $a_{n+1}=q\times a_n$
- Zatem jeśli np.
  - a=3, q=5, to mamy ciąg 3, 15, 75, 375, ...
  - a=-6, q=2, to mamy ciąg -6, 36, -216, 1496...
  - a=0, q=0, to mamy ciąg 0, 0, 0, ...
- Można też ogólnie podać wzór $a_n=a_0q^n$, (albo $a_n=a_1q^{n-1}$),co będzie stanowiło inną definicję.

Przykład nieco ciekawszy – Liczby Fibonacciego

- $F_0=0$,
- $F_1=1$,
- $F_n=F_{n-2}+F_{n-1}$ dla $n>1$.
- Mamy więc taki ciąg wartości:

0 1 1 2 3 5 8 13 21 34 55 89 144 233 377...

- Czy potrafimy podać wzór na n-tą liczbę Fibonacciego?

Nieco inne podejście

- Tym razem nieco zmienimy zasadę: to nie ja znając jakąś wartość rozglądam się, komu by ją oznajmić (tak jak a_n oznajmiał ją a_{n+1}), tylko odwrotnie – nie znając wartości rozglądam sie za tymi, której obliczenie mogą mi umożliwić.
- W zasadzie poprzednie podejście też by tak można było przedstawić (a_{n+1} pyta się , czy a_n może mu już pomóc).

Przykład

- $P_{i,j}$ – zestaw wartości dla różnych i,j takich, że $0 \leq i \leq j$.
  - $P_{i,j}=0$ dla każdych i,j takich, ze $i<j$
  - $P_{0,j}=1$, dla każdego j
  - $P_{i,j}=P_{i-1,j}+P_{i-1,j-1}$

Tabela dla $P_{i,j}$

- Pierwsza kolumna jest złożona z jedynek
- Pierwszy wiersz ma same zera poza pierwszym elementem.

1	0	0	0	0	0
1	1	0	0	0	0
1	2	1	0	0	0
1	3	3	1	0	0
1					
1					
1					

Tabela dla $P_{i,j}$

- Pierwsza kolumna jest złożona z jedynek
- Pierwszy wiersz ma same zera poza pierwszym elementem.

1	0	0	0	0	0
1	1	0	0	0	0
1	2	1	0	0	0
1	3	3	1	0	0
1	4				
1					
1					

Tabela dla $P_{i,j}$

- Pierwsza kolumna jest złożona z jedynek
- Pierwszy wiersz ma same zera poza pierwszym elementem.

1	0	0	0	0	0
1	1	0	0	0	0
1	2	1	0	0	0
1	3	3	1	0	0
1	4	6			
1					
1					

Tabela dla $P_{i,j}$

- Pierwsza kolumna jest złożona z jedynek
- Pierwszy wiersz ma same zera poza pierwszym elementem.

1	0	0	0	0	0
1	1	0	0	0	0
1	2	1	0	0	0
1	3	3	1	0	0
1	4	6	4		
1					
1					

Tabela dla $P_{i,j}$

- Pierwsza kolumna jest złożona z jedynek
- Pierwszy wiersz ma same zera poza pierwszym elementem.

1	0	0	0	0	0
1	1	0	0	0	0
1	2	1	0	0	0
1	3	3	1	0	0
1	4	6	4	1	0
1					
1					

Tabela dla $P_{i,j}$

- Pierwsza kolumna jest złożona z jedynek
- Pierwszy wiersz ma same zera poza pierwszym elementem.

1	0	0	0	0	0
1	1	0	0	0	0
1	2	1	0	0	0
1	3	3	1	0	0
1	4	6	4	1	0
1	5				
1					

Trójkąt Pascala

- Pierwsza kolumna jest złożona z jedynek
- Pierwszy wiersz ma same zera poza pierwszym elementem.

1	0	0	0	0	0
1	1	0	0	0	0
1	2	1	0	0	0
1	3	3	1	0	0
1	4	6	4	1	0
1	5	10	10	5	1
1	6	15	20	15	6

Wzór na $P_{i,j}$

- Czy uda się nam wyprowadzić wzór na $P_{i,j}$?
- Nie jest to proste, ale istnieje eudowodniony przez Pascala wzór korzystający z funkcji “silnia”.
- $n!=1\times2\times3\times4\times...\times n$
- ... albo po naszemu:
  - $0!=1$
  - $n!=n\times(n-1)!$

Liczba wyborów podzbiorów

- Wzór Pascala

$$P_{i,j} = \frac{i!}{j!(i-j)!}$$

Bardzo ciekawa funkcja

- A(m,n) – funkcja Ackermanna, określona dla naturalnych wartości m,n:
  - A(0,n)=n+1
  - A(m,0)=A(m-1,1) dla m>0
  - A(m,n)=A(m-1,A(m,n-1)) dla n,m>0

Wartości funkcji Ackermanna

- Początkowy fragment tabeli

	0	1	2	3	4	5	...	n
0	1	2	3	4	5	6	...	n+1
1	2	3	4	5	6	7	,,,	2+(n+3)-3
2	3	5	7	9	11	13	,,,	$2\times(n+3)-3$
3	5	13	29	61	125		...	$2^{(n+3)}-3$
4	13	65533	$2^{65536}-3$	$A(3,2^{65536}-3)$	$A(3,A(4,3))$		...	$2\uparrow 2\uparrow 2\uparrow \ldots \uparrow 2-3$
5								n+3 razy

Ciekawa własność funkcji Ackermanna

- Nie znamy wzoru na A(m,n). Jedyna dostępna metoda wyznaczenia wartości tej funkcji, to po kolei je obliczać korzystając z wcześniejszych wartości.
- Nie tylko nie znamy wzoru, ale wiadomo, że nie może go być!
- W pewnym sensie definicja indukcyjna okazała się silniejsza niż przez podanie wartości wprost!

Dowody indukcyjne

- Podobnie, jak definiowaliśmy różne pojęcia indukcyjnie, możemy dowodzić pewnych własności.
- Chodzi o dowodzenie zdań o liczbach naturalnych (na razie!), czy wręcz o ciągach wartości naturalnych.
- Idea jest ta sama: pokazujemy, że pewna własność zachodzi dla jakiejś początkowej wartości (np. 0), a potem pokazujemy, że jeśli zachodzi dla n, to zachodzi też dla n+1

Spróbujmy!

- Pokażmy, że dla ciągu arytmetycznego $a_1=a$, $a_{n+1}= a_n+d$ zachodzi wzór

$$a_n=a+(n-1)d$$

- Dla n=1 mamy $a_1= a+(1-1) =a$, czyli się zgadza.
- Załóżmy więc, że dla k=1,2,...,n wzór jest prawdziwy, czyli $a_k=a+(k-1)d$
- . Pokażemy, że dla k=n+1 też jest prawdziwy. Zatem $a_{n+1}= a_n + d = a+(n-1)d +d = a+nd-d+d=a+nd$ OK!

Coś trudniejszego!

- Pokażmy, że dla tego samego ciągu arytmetycznego $a_1+a_2+...+a_n = ½ (a_1+a_n)\times n$
- Zatem dla n=1 mamy $½ (a_1+a_1)\times 1 = a_1$, czyli się zgadza.
- Załóżmy więc, że dla k=1,2,...,n wzór jest prawdziwy. Pokażemy, że dla k=n+1 też jest prawdziwy. Zatem chcemy pokazać, że $a_1+a_2+...+a_n+a_{n+1}= ½ (a_1+a_{n+1})\times(n+1)$ korzystając z założenia, że $a_1+a_2+...+a_n = ½ (a_1+a_n)\times n$

Coś trudniejszego!

- $a_1+a_2+...+a_n+a_{n+1}=$
  - $= ½ (a_1+a_n)\times n + a_{n+1} =$
  - $= ½ (a_1+a_1+(n-1)d)\times n + a_1+nd=$
  - $= ½ (2na_1+n(n-1)d + 2a_1+2nd)=$
  - $= ½ (2(n+1)a_1+nd(n-1+2))=$
  - $= ½ (2(n+1)a_1+nd(n+1))=$
  - $= ½ (2a_1+nd)(n+1)=$
  - $= ½ (a_1+a_1+nd)(n+1)=$
  - $= ½ (a_1+a_{n+1})(n+1)$ CND

Jak się ma indukcja do programowania?

- Bardzo często rozwiązując problemy potrafimy sprowadzić je do rozwiązań dla danych mniejszych rozmiarów.
- Wystarczy zatem opracować ten jeden krok: jak z rozwiązania dla małych danych utworzyć rozwiązanie dla (nieco) większych danych plus rozwiązać problem dla bardzo małych danych!

Divide et impera

- Starożytni Rzymianie znali tę zasadę
- Łatwiej się rządzi, jeśli poddani są podzieleni
- Nie chodziło im jednak bynajmniej o podziały administracyjne
- Chodziło o to, zeby skłócić poddanych!
- ... my zrobimy to bardziej humanitarnie.

Zasada dziel i rządź w programowaniu

- Chodzi o to, żeby po pierwsze umieć rozwiązać problem dla małych danych
- ... a po drugie, żeby mając duże dane tak je podzielić, aby problem sprowadzić do kilku mniejszych podproblemów, a potem scalić wyniki.
- Idealnie nadaje się do tego mechanizm rekursji, ale można sie często bez niej obyć.

Znajdowanie minimum i maksimum w tablicy

- Jak znaleźć jednocześnie dwie wartości: największą i najmniejszą w tablicy?
- Rozwiązanie:
  - podzielmy tablicę na dwie „połówki”
  - rozwiążmy problem maksimum-minimum w każdej z nich, otrzymując pary (minl,maxl),(minp,maxp)
  - wynik, to para(min(minl,minp), max(maxl,maxp))
- Czy coś zyskaliśmy?

Analiza złożoności

- Gdybyśmy te dwie wartości znajdowali po kolei, wykonalibyśmy 2(n-1) porównań.
- Oznaczmy przez T(n) liczbę porównań wykonywanych przez nasz algorytm dla n danych
  - T(1)=0
  - T(n)=2T(n/2)+2
- Rozwiązanie tego układu jest dość proste. Można przez indukcję pokazać, że dla $n=2^k$
  - T(n)=(3/2)n-2

Twierdzenie Pohla

- Problemu min-max nie da się rozwiązać taniej niż za pomocą ⌜(3/2)n⌝-2 porównań.
- Dowód (bardzo ciekawy) w książce L.Banachowski, A.Kreczmar „Elementy Analizy Algorytmów”.

<Mistrzostwa
w Algorytmice
i Programowaniu>

Projekt „Mistrzostwa w Algorytmice i Programowaniu – Uczniowie” jest finansowany ze środków pochodzących z „Programu Rozwoju Talentów Informatycznych na lata 2019-2029”

Dofinansowanie Projektu: 4.887.850,50 zł

Całkowita wartość Projektu: 5.460.850,50 zł