

# TEORETYCZNE PODSTAWY INFORMATYKI

14/12/2015

WFAiS UJ, Informatyka Stosowana  
I rok studiów, I stopień

# Wykład 9 – część I

2

## Kombinatoryka

- Wariacje z powtórzeniami
- Permutacje
- Wariacje bez powtórzeń
- Kombinacje
- Łączenie reguł kombinatorycznych

# Kombinatoryka i prawdopodobieństwo

3

- Często spotykamy się z problemem obliczenia wartości wyrażającej prawdopodobieństwo zajścia określonych zdarzeń.
- Dziedzina matematyki zajmująca się tą tematyką to **kombinatoryka**.
- Pojęcia związane z próbami szacowania prawdopodobieństwa występowania zdarzeń definiuje **teoria prawdopodobieństwa**.
  
- Zaczniemy od **kombinatoryki...**

# Wariacje z powtórzeniami

4

- Jednym z najprostszycch, ale też najważniejszych problemów jest analiza listy elementów, z których każdemu należy przypisać jedną z wartości należących do stałego zbioru.
- Należy określić możliwą liczbę różnych przyporządkowań (wariacji z powtórzeniami) wartości do elementów.
- Przykład:  
4 kwadraty, każdy można pokolorować jednym z 3 kolorów.  
Ile możliwych pokolorowań?  $3 \cdot 3 \cdot 3 \cdot 3 = 3^4 = 81$



# Wariacje z powtórzeniami

5

- Mamy listę  $n$ -elementów. Istnieje zbiór  $k$ -wartości z których każda może być przyporządkowana do jakiegoś elementu. Przyporządkowanie jest listą  $n$  wartości  $(n_1, n_2, \dots, n_n)$ . Gdzie każda z  $n_1, n_2, \dots, n_n$  jest jedną z wartości  $k$ .
  - ▣ Istnieje  $k^n$  różnych przyporządkowań.
- **Twierdzenie:**  
 **$S(n)$ : liczba możliwych sposobów przyporządkowania dowolnej z  $k$  wartości do każdego z  $n$  elementów wynosi  $k^n$ .**

# Wariacje z powtórzeniami

6

## □ Podstawa:

Przypadek podstawowy to  $n=1$ . Jeżeli mamy 1 element możemy wybrać dla niego dowolną spośród  $k$  wartości. Istnieje więc  $k$  różnych przyporządkowań. Ponieważ  $k^1=k$ , podstawa indukcji jest prawdziwa.

## □ Indukcja:

Założmy że  $S(n)$  jest prawdziwe i rozważmy  $S(n+1)$ , określając że istnieje  $k^{n+1}$  możliwych przyporządkowań jednej z  $k$  wartości do każdego z  $n+1$  elementów.

- Wiemy, że istnieje  $k$  możliwości doboru wartości dla pierwszego elementu. Zgodnie z hipotezą indukcyjną, istnieje  $k^n$  przyporządkowań wartości do pozostałych  $n$  elementów. Łączna liczba przyporządkowań wynosi  $k \cdot k^n = k^{n+1}$ . Cnd.

# Permutacje

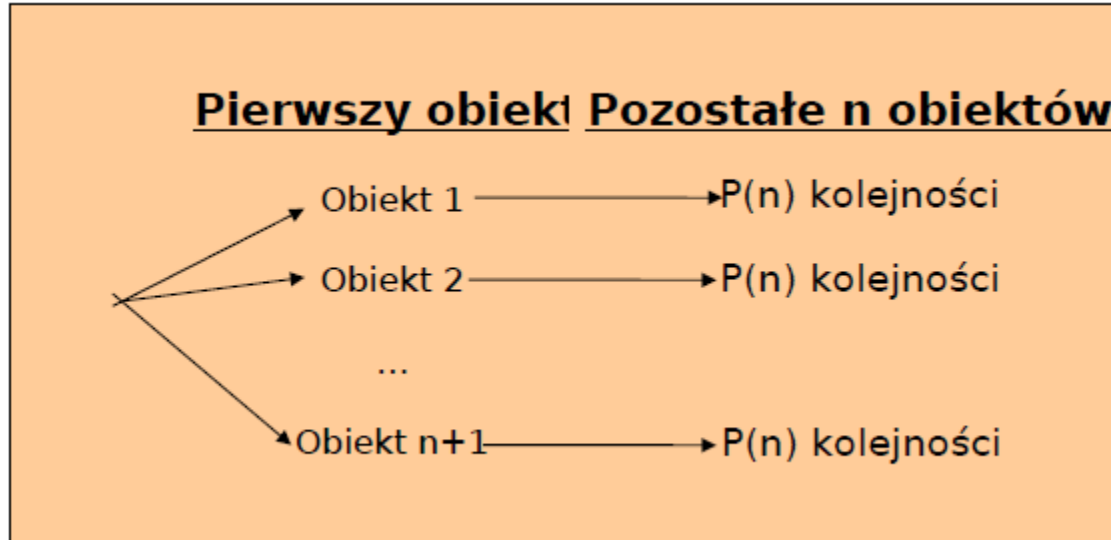
7

- Mając  $n$  różnych obiektów, na ile różnych sposobów można je ustawić w jednej linii?
  - Każde takie uporządkowanie nazywamy permutacją.
  - Liczbę permutacji  $n$  obiektów zapisujemy jako  $P(n)$ .

# Jak obliczyć $P(n+1)$ ?

8

- Problem: mamy  $n+1$  obiektów ( $a_1, a_2, \dots, a_n, a_{n+1}$ ) które mają zostać ustawione.
- Ilość możliwych wyników ustawień jest  $P(n+1)$



Permutacje  
 $n+1$  obiektów



# Jak obliczyć $P(n+1)$ ?

9

## □ Twierdzenie:

$P(n) = n!$  dla wszystkich  $n \geq 1$

## □ Podstawa:

Dla  $n=1$ ,  $P(1)=1$  określa że istnieje jedna permutacja dla jednego obiektu.

## □ Indukcja:

- Załóżmy że  $P(n) = n!$
- Wówczas wg. naszego twierdzenia:  $P(n+1)=(n+1)!$
- Rozpoczynamy od stwierdzenia że  $P(n+1)=(n+1) \cdot P(n)$
- Zgodnie z hipotezą indukcyjną  $P(n)=n!$ , zatem  $P(n+1)=(n+1) \cdot n!$
- Zatem  $P(n+1)=(n+1) \cdot n! = (n+1) \cdot n \cdot (n-1) \cdot (n-2) \cdot \dots \cdot 1 = (n+1)!$ , czyli nasze twierdzenie jest poprawne. Cnd.

# Jak obliczyć $P(n+1)$ ?

10

Jednym z interesujących zastosowań wzoru na liczbę permutacji jest dowód na to że algorytmy sortujące muszą działać w czasie co najmniej proporcjonalnym do  $(n \log n)$ , dla  $n$  elementów do posortowania, chyba że wykorzystują jakieś specjalne własności sortowanych elementów.

# Wariacje bez powtórzeń

11

- Niekiedy chcemy wybrać tylko niektóre spośród elementów zbioru i nadać im określony porządek.
- Uogólniamy opisaną poprzednio funkcję  $P(n)$  reprezentującą liczbę permutacji, aby otrzymać **dwuargumentową funkcję  $P(n,m)$** , którą definiujemy jako ilość możliwych sposobów wybrania  **$m$  elementów z  $n$ -elementowego zbioru**, przy czym **istotną rolę** odgrywa **kolejność wybierania elementów**, natomiast nieważne jest uporządkowanie elementów nie wybranych.
- Zatem  **$P(n) = P(n,n)$** .

# Wariacje bez powtórzeń

12

## Przykład:

Ile istnieje sposobów utworzenia sekwencji **m** liter ze zbioru **n** liter, jeżeli żadna litera nie może występować więcej niż raz?

- Warunek aby zadanie miało sens:  **$n \geq m$**
- Pierwszą literę możemy wybrać na **n sposobów** (wybieramy ze zbioru n-elementowego), drugą **na n-1 sposobów** (gdyż nie możemy wybrać tej samej litery co poprzednio), trzecia na **n-2 sposoby**, ...
- Ostatnią na **n-(m-1) sposobów**.

Twierdzenie:  $P(n,m) = n \cdot (n-1) \cdot \dots \cdot (n-(m-1))$  dla wszystkich  $m \leq n$

Twierdzenie:  $P(n,m) = \frac{n!}{(n-m)!}$  dla wszystkich  $m \leq n$

# Kombinacje

13

- Kombinacja to każdy podzbiór zbioru skończonego.
- Kombinacja  $m$ -elementową zbioru  $n$ -elementowego  $A$  nazywa się każdy  $m$ -elementowy podzbiór zbioru  $A$  ( $0 \leq m \leq n$ ). Używa się terminu „kombinacja z  $n$  elementów po  $m$  elementach” lub wręcz „kombinacja z  $n$  po  $m$ ”.
- Taką funkcję zapisujemy jako:

$$\binom{n}{m} = \frac{P(n,m)}{P(m)} = \frac{n!}{(n-m)! \cdot m!}$$

# Wyznaczanie liczby kombinacji

14

**Podstawa:**  $\binom{n}{0} = 1$  dla dowolnego  $n \geq 1$ .

Oznacza to że istnieje tylko jeden sposób wybrania **zero** elementów ze zbioru **n**-elementowego – wybranie niczego.

Także  $\binom{n}{n} = 1$ , ponieważ jedynym sposobem wybrania **n**-elementów ze zbioru **n**-elementowego jest wybranie ich wszystkich.

**Indukcja:** Jeśli  $0 < m < n$ , to  $\binom{n}{m} = \binom{n-1}{m} + \binom{n-1}{m-1}$ .

Oznacza to, że jeżeli chcemy wybrać **m** elementów ze zbioru **n**-elementowego, możemy albo:

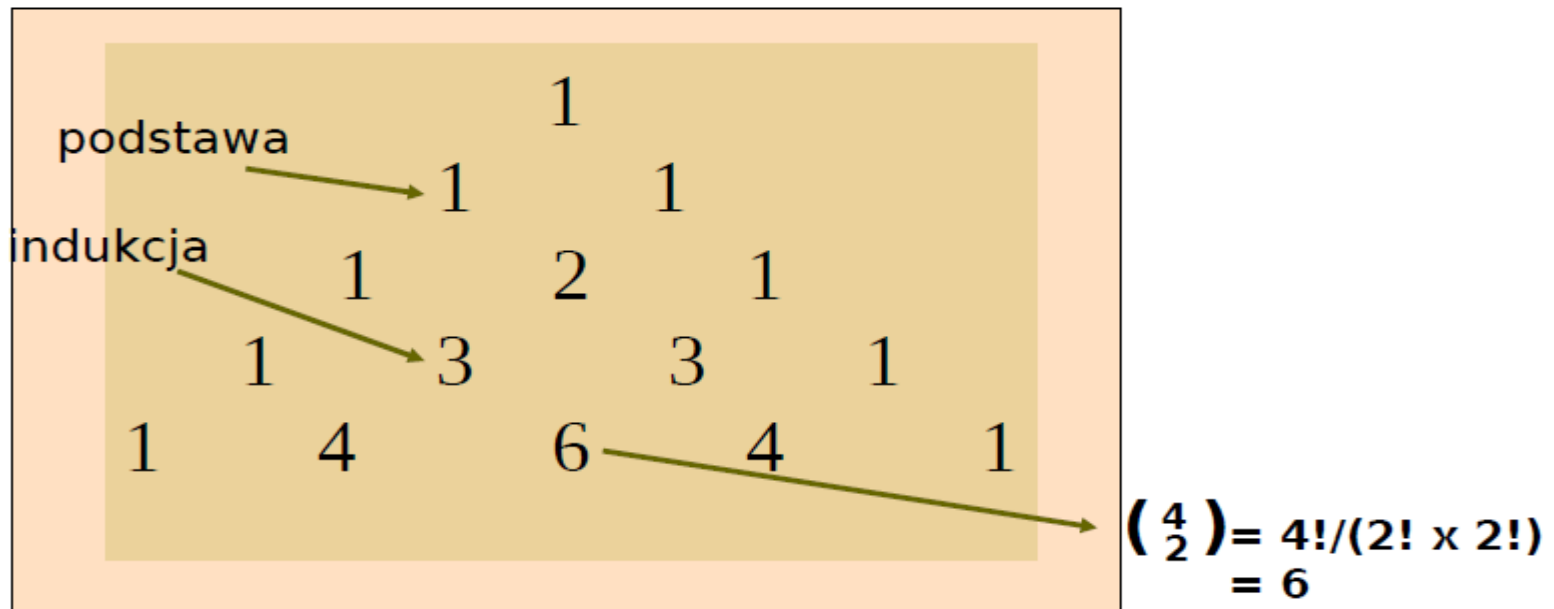
- nie wybrać pierwszego elementu, po czym wybrać **m** elementów z pozostałych **n-1** elementów. Taką liczbę możliwości wyraża  $\binom{n-1}{m}$ .
- wybrać pierwszy element, po czym wybrać **m-1** elementów z pozostałych **n-1** elementów. Taką liczbę możliwości wyraża  $\binom{n-1}{m-1}$ .

# Trójkąt Pascala

15

- Rekurencję przy obliczaniu liczby kombinacji często ilustruje się przy pomocy trójkąta Pascala.

$\binom{n}{m}$  =  $(m+1)$  liczba w  $(n+1)$  wierszu



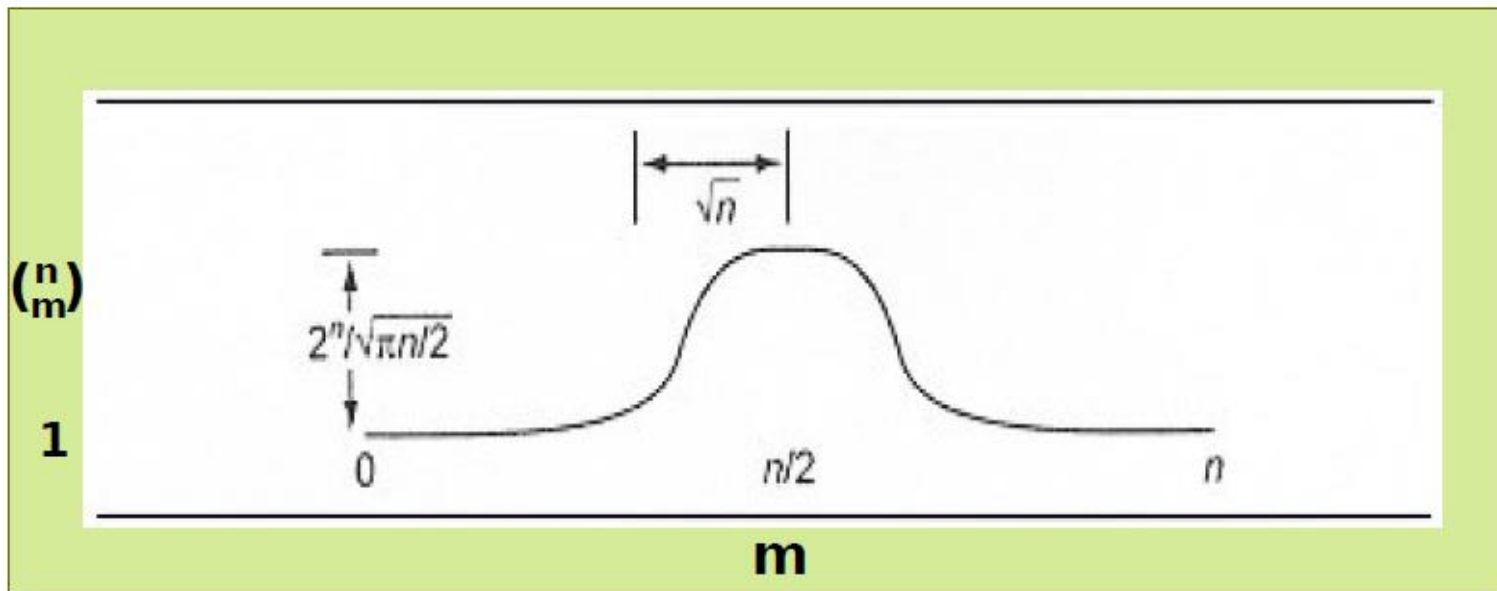
# Interesujące własności funkcji $\binom{n}{m}$

16

- To są również współczynniki rozkładu dwuwyrzowego wielomianu (dwumianu)  $(x+y)^n$

- $\sum_{m=0}^n \binom{n}{m} = 2^n$

- Wykres funkcji  $\binom{n}{m}$  dla stałej dużej wartości  $n$ :





# Łączenie reguł kombinatorycznych

17

- Typowy problem kombinatoryczny wymaga **łączenia przedstawionych reguł** (cegiełek) w bardziej skomplikowane struktury.
- Techniki których używamy to:
  - ▣ prowadzenie obliczeń jako **sekwencji wyborów**;
  - ▣ prowadzenie obliczeń jako **różnicy** innych obliczeń (np. wszystkich wyborów – nieprawidłowych wyborów );
  - ▣ prowadzenie obliczeń jako **sumy rozwiązań** dla podzypadków które są wzajemnie rozłączne.

# Wykład 9 – część II

18

Prawdopodobieństwo i algorytmy probabilistyczne

- Co to jest teoria prawdopodobieństwa
- Podstawowe pojęcia:
  - ▣ reguła sum, reguła iloczynów
  - ▣ prawdopodobieństwa warunkowe
- Przykład z kartami
- Analiza probabilistyczna
- Liczby losowe, generatory liczb losowych
- Algorytmy wykorzystujące prawdopodobieństwo
  - ▣ Czy pudełko jest wadliwe?
  - ▣ Czy liczba  $N$  jest liczbą pierwszą?

# Teoria prawdopodobieństwa

19

- Teoria prawdopodobieństwa, szeroko stosowana we współczesnej nauce, ma również wiele zastosowań w informatyce, np.:
  - szacowanie czasu działania programów dla przypadków ze średnimi, czyli typowymi danymi wejściowymi,
  - wykorzystanie do projektowania algorytmów „podejmujących decyzje” w niepewnych sytuacjach, np. najlepsza możliwa diagnoza medyczna na podstawie dostępnej informacji,
  - algorytmy typu Monte Carlo,
  - różnego rodzaju symulatory procesów,
  - prawie zawsze „prawdziwe” rozwiązania.

# Podstawowe pojęcia

20

## Przestrzeń probabilistyczna $W$ :

- Skończony zbiór punktów, z których każdy reprezentuje jeden z możliwych wyników doświadczenia. Każdy punkt  $x$  jest związany z taką nieujemną liczbą rzeczywistą zwaną prawdopodobieństwem  $x$ , że suma prawdopodobieństw wszystkich punktów wynosi 1. Istnieje także pojęcie nieskończonych przestrzeni probabilistycznych ale nie mają one większego zastosowania w informatyce.

## Zdarzenie $E$ :

- Podzbiór punktów w przestrzeni probabilistycznej. Prawdopodobieństwo zdarzenia,  $P(E)$ , jest sumą prawdopodobieństw punktów należących do tego zdarzenia.

## Dopełnienie zdarzenia $E$ czyli $\bar{E}$ :

- Zbiór punktów przestrzeni probabilistycznej które nie należą do zdarzenia  $E$ .  $P(E) + P(\bar{E}) = 1$ .

# Prawdopodobieństwo warunkowe

21

- Prawdopodobieństwem warunkowym zajścia zdarzenia  $F$  pod warunkiem zajścia zdarzenia  $E$ , gdzie  $P(E) > 0$  nazywamy liczbę:  
$$P(F|E) = P(E \cap F) / P(E).$$
- Jest to iloraz prawdopodobieństwa części wspólnej zdarzeń  $E, F$  i prawdopodobieństwa zdarzenia  $E$ .
- Zdarzenia  $E, F$  nazywamy niezależnymi jeśli zachodzi:  
$$P(E \cap F) = P(E) \cdot P(F)$$
- W przeciwnym wypadku zdarzenia są zależne.
- Dla zdarzeń niezależnych  $E, F$  zachodzi:  
$$P(F|E) = P(F)$$

# Przykłady

22

## Zdarzenia niezależne:

- Rzucamy dwoma kostkami, wyrzucenie liczby „1” na pierwszej kostce (zdarzenie E) nie wpływa na możliwość pojawienia się liczby „1” na drugiej kostce (zdarzenie F).  $P(F|E) = P(F)$

## Zdarzenia zależne:

- Ciągniemy dwa razy kartę z talii kart. Wyciągnięcie jako pierwszej karty asa (zdarzenie E), wpływa na możliwość wyciągnięcia jako drugiej karty asa (zdarzenie F).  $P(F|E) \neq P(F)$ .

W niektórych sytuacjach liczenie prawdopodobieństw jest łatwiejsze jeżeli podzielimy przestrzeń probabilistyczna na rozdzielne obszary  $R_1, R_2, \dots, R_k$ . Wówczas  $P(E) = \sum_{i=1}^k P(E|R_i)P(R_i)$ .

# Przykład z kartami

23

- Ciągniemy dwie karty z talii 52 kart. Liczba możliwych wyników tego doświadczenia (czyli wariacji bez powtórzeń) wynosi  $|W| = 52 * 51 = 2652$ .
- Oznaczmy poprzez E zdarzenie polegające na wyciągnięciu jako pierwszej karty As'a.  
 $|E| = 4 * 51 = 204$ .  
 $P(E) = |E|/|W| = 204/2652 = 1/13$ .
- Prawdopodobieństwo wyciągnięcia As'a jako drugiej karty, (zdarzenie F), jeżeli pierwsza wyciągnięta karta to był As jest
  - ▣  $P(F|E) = P(E \cap F) / P(E) = 4 \cdot 3 / 204 = 1/17$ .
- **$P(E \cap F) = P(E) \cdot P(F|E) = 1/13 \cdot 1/17 = 1/221$ .**

# Przykład z kartami

24

- Podzielmy przestrzeń na dwa obszary:
  - $R_1$  - pierwszą kartą jest As,  $|R_1| = 4 \cdot 51 = 204$ .
  - $R_2$  - pierwszą kartą nie jest As,  $|R_2| = 2652 - 204 = 2448$ .
  - $P(E \cap F) = P((E \cap F)|R_1) \cdot P(R_1) + P((E \cap F)|R_2) \cdot P(R_2)$
  - $P(E \cap F | R_2) = 0$
  - $P(E \cap F | R_1) = 4 \cdot \frac{3}{4} \cdot 51 = 1/17$
  - $P(R_1) = 204 / 2652 = 1/13$
  - **$P(E \cap F) = P((E \cap F)|R_1) \cdot P(R_1) + P((E \cap F)|R_2) \cdot P(R_2)$**   
 $= 1/17 \cdot 1/13 + 0 \cdot P(R_2)$   
 **$= 1/221$ .**
  
- Gdybyśmy po wyciągnięciu pierwszej karty zwracali ją z powrotem do talii, to mielibyśmy  $P(F|E) = P(F) = 1/13$  (zdarzenia niezależne).
  - Wówczas  $P(E) \cdot P(F|E) = 1/13 \cdot 1/13 = 1/169$ .



# Reguły związane z wieloma zdarzeniami

25

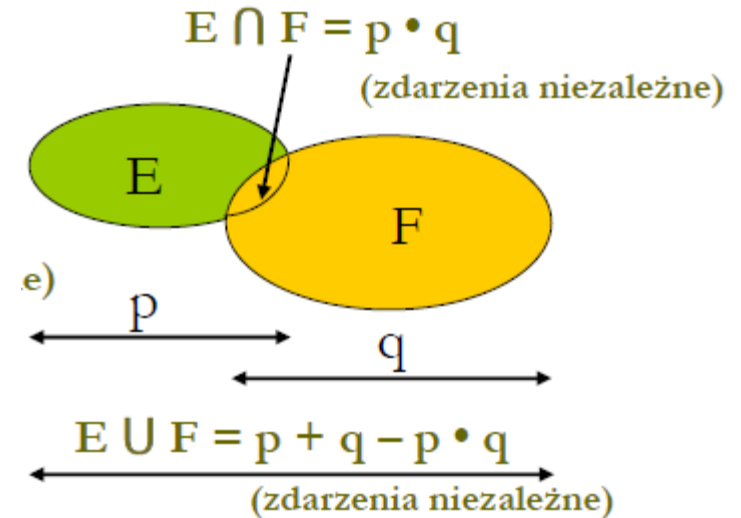
Oznaczmy:  $P(E) = p$ ,  $P(F) = q$ .

Wówczas:

$$\max(0, (p+q-1)) \leq P(E \cap F) \leq \min(p, q)$$

$$P(E \cup F) = p + q - P(E \cap F)$$

$$P(E \cup F) = p + q - p \cdot q \text{ (zdarzenia niezależne)}$$



W zastosowaniach czasem akceptujemy że nie możemy wyznaczyć dokładnie prawdopodobieństw oraz zależności między zdarzeniami. Potrafimy tylko wskazać sytuacje najmniej lub najbardziej prawdopodobne.

**Zastosowanie: różnego typu diagnostyka**

# Oczekiwane wartości obliczeń i analiza probabilistyczna

26

- Przypuśćmy, że mamy pewną funkcję określoną na przestrzeni probabilistycznej  $f(x)$ . Wartość oczekiwana tej funkcji po wszystkich punktach przestrzeni

$$E(f) = \sum f(x) P(x)$$

- Mamy tablicę  $n$  liczb całkowitych, sprawdzamy czy jakaś liczba całkowita „ $x$ ” jest elementem tej tablicy. Algorytm przegląda całą tablicę, po napotkaniu

$A[i] = x$  kończy działanie.

- Jeżeli  $A[0] = x$  to algorytm  $O(1)$
- Jeżeli  $A[n-1] = x$  to algorytm  $O(n)$
- $E(f) = \sum_{i=0}^{n-1} (ci + d) \cdot (1/n) = c \cdot (n-1) / 2 + d$
- $E(f) \sim c \cdot n/2$  dla dużego  $n$

1	$A[0]$
8	
7	
5	
3	
4	$A[i]$
8	
9	
7	$A[n-1]$

# Algorytmy wykorzystujące prawdopodobieństwo

27

- Jest bardzo wiele różnych typów algorytmów wykorzystujących prawdopodobieństwo.
- Jeden z nich to tzw. **algorytmy Monte-Carlo** które wykorzystują liczby losowe do zwracania albo wyniku pożądanego („prawda”), albo żadnego („nie wiem”).
  - Wykonując algorytm **stałą** liczbę razy, możemy rozwiązać problem, dochodząc do wniosku, że jeśli żadne z tych powtórzeń nie doprowadziło nas do odpowiedzi „prawda”, to odpowiedzią jest „fałsz”.
  - Odpowiednio dobierając liczbę powtórzeń, możemy dostosować prawdopodobieństwo niepoprawnego wniosku „fałsz” do tak niskiego poziomu, jak w danym przypadku uznamy za konieczne.
  - **Nigdy** jednak nie osiągniemy prawdopodobieństwa popełnienia błędu na poziomie **zero**.

# Co to są liczby losowe?

28

- Mówimy, że wyniki pewnych doświadczeń są **losowe**, co oznacza że wszystkie możliwe **wyniki są równie prawdopodobne**.
- Przykładowo, jeżeli rzucamy normalną (prawkidlową) kostką do gry to zakładamy że nie ma możliwości fizycznego kontrolowania wyniku tego rzutu w taki sposób aby jeden wynik był bardziej prawdopodobny od drugiego.
- Podobnie zakładamy że mając uczciwie potasowaną talie kart nie możemy wpłynąć na wynik - prawdopodobieństwo otrzymania w rozdaniu każdej karty jest identyczne.

# Generatory liczby losowych?

29

- Wszystkie generowane przez komputer losowe sekwencje są wynikiem działania specjalnego rodzaju algorytmu zwanego generatorem liczb losowych (ang. random number generator).
- Przykład prostego generatora który całkiem dobrze sprawdza się w praktyce to tzw. “liniowy generator kongurencyjny”.
  - Wyznaczamy stałe  $a \geq 2$ ,  $b \geq 1$ ,  $x_0 \geq 0$  oraz współczynnik  $m > \max(a, b, x_0)$ . Możemy teraz wygenerować sekwencje liczb  $x_1, x_2, \dots$  za pomocą wzoru:  
$$x_{n+1} = (a x_n + b) \bmod(m)$$
  - Dla właściwych wartości stałych  $a, b, m$  oraz  $x_0$ , sekwencja wynikowa będzie wyglądała na losową, mimo że została ona wygenerowana przy użyciu konkretnego algorytmu i na podstawie “jądra”  $x_0$ .
- Dla szeregu zastosowań istotna jest odtwarzalność sekwencji liczb losowych.

# Algorytmy wykorzystujące prawdopodobieństwo

30

- Mamy pudełko w którym jest  $n$ -procesorów, nie mamy pewności czy zostały przetestowane przez producenta. Zakładamy że prawdopodobieństwo że procesor jest wadliwy (w nieprzetestowanym pudełku) jest 0.10.
- **Co możemy zrobić aby potwierdzić czy pudełko jest dobre?**
  - ▣ przejrzeć wszystkie procesory  $\rightarrow$  algorytm  $O(n)$
  - ▣ losowo wybrać  $k$  procesorów do sprawdzenia  $\rightarrow$  algorytm  $O(1)$
  - ▣ błąd polegałby na uznaniu że pudełko dobre (przetestowane) jeżeli nie było takie.
- Losujemy  $k=131$  procesorów.  
Jeżeli procesor jest dobry odpowiadamy „nie wiem”. Prawdopodobieństwo że „nie wiem” dla każdego z  $k$ -procesorów  $(0.9)^k = (0.9)^{131} = 10^{-6}$ .  
 $10^{-6}$  to jest prawdopodobieństwo że pudełko uznamy za dobre choć nie było testowane przez producenta.  
Za cenę błędu  $= 10^{-6}$ , zamieniliśmy algorytm z  $O(n)$  na  $O(1)$ .  
Możemy regulować wielkość błędu/czas działania algorytmu zmieniając  $k$ .

# „ Czy liczba $N$ jest liczbą pierwszą ?”

31

- W połowie lat 70-tych odkryto **dwa bardzo eleganckie probabilistyczne algorytmy** sprawdzające, czy liczba jest pierwsza. Były one jednymi z pierwszych rozwiązań probabilistycznych dla trudnych problemów algorytmicznych. Wywołały fale badań które doprowadziły do probabilistycznych rozwiązań wielu innych problemów.
- Oba algorytmy wykonują się w czasie wielomianowym (niskiego stopnia), zależnym od liczby cyfr w danej liczbie  $N$  (czyli  $O(\log N)$ ).
- Oba algorytmy są oparte na losowym szukaniu pewnych rodzajów potwierdzeń lub **świadcstw złożoności** liczby  $N$ .
- Po znalezieniu takiego świadectwa algorytm może się bezpiecznie zatrzymać z odpowiedzią „**nie,  $N$  nie jest liczbą pierwszą**”, ponieważ istnieje bezdyskusyjny dowód że  $N$  jest liczbą złożoną.
- Poszukiwanie musi być przeprowadzone w taki sposób aby w pewnym rozsądnym czasie algorytm mógł przerwać szukanie odpowiadając, że  $N$  jest liczbą pierwszą z bardzo małą szansą omyłki.
- Trzeba zatem znaleźć dająca się **szybko sprawdzać** definicje świadectwa złożoności.

# „ Czy liczba N jest liczbą pierwszą ?”

32



jeśli to jest odpowiedź to jest to prawda z błędem  $1/2^{200}$

jeśli to jest odpowiedź to jest to prawda



# Świadectwa złożoności

33

- Każda liczba parzysta oprócz 2 jest złożona
- Jeżeli suma cyfr liczby jest podzielna przez 3 to liczba jest złożona (iteracyjny prosty algorytm liniowo zależny od liczby cyfr)
- Test pierwszości Fermata:
  - jeśli  $n$  jest liczbą pierwszą oraz  $k$  jest dowolna liczba całkowita  $(1, n-1)$ , to  $k^{n-1} \equiv 1 \pmod{n}$ .
  - natomiast jeśli  $n$  jest liczbą złożoną (z wyjątkiem kilku złych liczb złożonych – liczb Carmichael’a) oraz jeśli  $k$  wybierzemy losowo z przedziału  $(1, n-1)$  to prawdopodobieństwo tego że  $k^{n-1} \not\equiv 1 \pmod{n}$  jest mniejsze niż  $\frac{1}{2}$ .
  - Zatem liczby złożone (poza liczbami Carmichael’a) spełniają warunek testu dla danego  $k$  z prawdopodobieństwem nie mniejszym niż  $\frac{1}{2}$ .
- Test pierwszości Solovay-Strassena:
  - jeśli  $k$  i  $n$  nie mają wspólnych dzielników (co by było świadectwem złożoności) policz:  
 $X = k^{(n-1)/2} \pmod{n}$ ,  $Y = J_s(n,k)$  (symbol Jacobiego),  
jeśli  $X \neq Y$  to  $k$  jest świadectwem złożoności liczby  $n$ .
  - dla tego testu nie ma ‘złych’ liczb złożonych.

# Podsumowanie

34

- **Przestrzeń probabilistyczna** składa się z punktów z których każdy reprezentuje wynik jakiegoś doświadczenia. Każdy punkt  $x$  związany jest z nieujemną liczbą zwaną prawdopodobieństwem punktu  $x$ . Suma prawdopodobieństw wszystkich punktów składających się na przestrzeń probabilistyczna wynosi 1.
- **Zdarzenie** jest podzbiorem punktów z przestrzeni probabilistycznej. Prawdopodobieństwo zdarzenia jest sumą prawdopodobieństw należących do niego punktów. Prawdopodobieństwo każdego zdarzenia mieści się w przedziale od 0 do 1.

# Podsumowanie

35

- **Reguła sum** określa, że prawdopodobieństwo tego, że zajdzie jedno z dwóch zdarzeń  $E$  lub  $F$  jest większe lub równe większemu z prawdopodobieństw obu zdarzeń, ale nie większa niż suma tych prawdopodobieństw.
- **Reguła iloczynów** określa, że prawdopodobieństwo tego, że wynikiem pewnego doświadczenia będą dwa zdarzenia  $E$  i  $F$ , jest nie większe niż mniejsze z prawdopodobieństw obu zdarzeń.
- Wykonując **algorytm Monte Carlo** stałą liczbę razy, możemy rozwiązać problem, dochodząc do wniosku, że jeśli żadne z tych powtórzeń nie doprowadziło nas do odpowiedzi „prawda”, to odpowiedzią jest „fałsz”.