6 Counting

6.1 The Basics of Counting

Product and Sum Rules

Suppose two task,

m number of ways to do task 1

n number of ways to do task 2

Do ***both*** *task 1* ***and*** *task 2:* ***Product Rule***

mn ways

$|A \times B| = |A| \cdot |B|$.

Do ***either*** *task 1* ***or*** *task 2, but not both:* ***Sum Rule***

m+n ways

$|A \cup B| = |A| + |B|$, *if* $A \cap B = \emptyset$.

Principle of Inclusion-Exclusion

$|A \cup B| = |A| + |B| - |A \cap B|$.

6.2 The Pigeonhole Principle

Theorem** 1 **The pigeonhole principle

If k + 1 or more objects are placed into k box,

there is at least one or more box containing two or more objects.

Dirichlet drawer principle

***Corollary** 1 If f: A → B and |A| ≥ |B|+1, then f is **not** one-to-one.*

*If f: A → B and |A| > |B|, then f is **not** one-to-one.*

Theorem** 2 **The generalized pigeonhole principle

If N objects are placed into k box,

there is at least one or more box containing $\lceil N/k \rceil$ *objects.*

***proof** by **contradiction**, no box contains* $\lceil N/k \rceil$ *objects.*

$k(\lceil N/k \rceil - 1) < k((N/k+1) - 1) = N$

***Example** N = 59 students, k = 12 months,*

$\lceil N/k \rceil = \lceil 4.91... \rceil = 5$ *students/month*

6.3 Permutations and Combinations

Thm. 1 $1 \le r \le n, P(n, r) = n(n-1) \ldots (n-r-1)$

Col. 1 $0 \le r \le n, P(n, r) = n! / (n-r)!$

Thm.2 $0 \le r \le n, C(n, r) = P(n, r) / P(r, r) = n! / r!(n-r)!$

$C(n, r) = C(n, n-r)$

6.4 Binomial Coefficient

$(x+y)^n = C(n, 0)x^n + C(n, 1)x^{n-1}y + \ldots + C(n, n-1)x^1y^{n-1} + C(n, n)y^{n-1}.$

$= \sum_{j=0}^{n} C(n, n-j)x^{n-j}y^j.$

Corollary 1 $\sum_{k=0}^{n} C(n, k) = 2^n$. for n

Corollary 2 $\sum_{k=0}^{n} (-1)^k C(n, k) = 2^n.$

Pascal's Identity and Trialgle

C(n+1, k) C(n, k-1) +

6.5 Generalized Permutation and Combinations

6.6 Generating Permutation and Combinations

Let $a_1a_2\ldots a_n$ *and* $b_1b_2\ldots b_n$ *be* **permutation** *of* $\{1, 2, \ldots, n\}$.

We say $a_1a_2\ldots a_n < b_1b_2\ldots b_n$,

if $1 \le \exists k \le n$.э. $a_1 = b_1, a_2 = b_2, \ldots, a_{k-1} = b_{k-1}, a_k < b_k$.

permutation $a_1a_2\ldots a_n \in \{1, \ldots, n\}^n$.

< is a total order

Next permutation of $a_1 \ldots a_n$?

Find largest j .∍. $a_j < a_{j+1}$.

Find smallest interger a_k in $a_{j+1}, \ldots, a_n$ but $a_k > a_j$.

Interchange a_k and a_j.

Increasing order from j+1-th to n-th position.

procedure *nextPermutation($a_1 \ldots a_n \in \{1, \ldots, n\}^n \neq n\ (n-1) \ldots 2\ 1$)*

j := n-1; ***do*** *$a_j > a_{j+1} \rightarrow j := j - 1$* ***od***

k:=n; ***do*** *$a_j > a_k \rightarrow k := k - 1$* ***od***

$a_k, a_j := a_j, a_k$;

r, s := n, j+1; ***do*** *$r > s \rightarrow a_r, a_s := a_s, a_r$; r, s:= r - 1, s+1* ***od***

procedure *next bit string*($b_1 \ldots b_n \in \{0, 1\}^n \neq 111 \ldots 11$)

$i := n$; **do** $b_i = 1 \rightarrow b_i := 0; i := i - 1$ **od**; $b_i := 1$.

Next r-combination of $\{a_1, a_2, \ldots, a_n\}$ *with* $a_1 < a_2 < \ldots < a_n$?

Find a_i .∍. $a_i \neq n - r + i$.

replace a_i *with* $a_i + 1$.

repalace a_j*'s with* $a_i + j - i$.

procedure *next r-comnination*($\{a_1, a_2, \ldots, a_n\} \subseteq \{1, 2, \ldots, n\}$

$\neq \{n\text{-}r+1, n\text{-}r+2, \ldots, n\}$ *and* $a_1 < a_2 < \ldots < a_n$)

$i := r$; **do** $a_i = n - r + i \rightarrow i := i - 1$ **od**; $a_i := a_i + 1$;

for $j:=i+1$ ***to*** r ***do*** $a_j := a_i + j - i$ ***od***