Probleme informatice (parțial) rezolvate matematic

2018

FMI - Lecții de pregătire pentru admitere

Scop

Trucuri (matematice) pentru a rezolva unele probleme informatice.

Scop

Trucuri (matematice) pentru a rezolva unele probleme informatice.
Trucuri (informatice) pentru a rezolva unele probleme matematice.
Provocare: Project Euler

Pe scurt

1. Probleme de numărare și Programare dinamică
2. Probleme cu numere; Divizibilitate; Algoritmul lui Euclid

Probleme de numărare: Plan

- Descoperirea formulei recursive
- (Opțional) rezolvarea recursiei
- Programarea soluției.

Problemă

Q1

Dacă o echipă de baschet are 9 jucători,
în câte moduri își poate alege cei 5 jucători în teren?

Truc matematic: Combinări de n luate câte k

Formula recursivă

$$C_n^0 \;=\; 1, \quad C_n^n \;=\; 1, \quad C_n^k \;=\; C_{n-1}^k + C_{n-1}^{k-1}$$

Rezolvarea recursiei

$$C_n^k = \frac{n!}{(n-k)!k!}$$

C_n^k — formula directă

$$C_n^k = C_n^{n-k} = \frac{n!}{(n-k)!k!}$$

C_n^k — formula directă

$$C_n^k = C_n^{n-k} = \frac{n!}{(n-k)!k!} = \frac{(n-k+1)\cdot(n-k+2)\cdot\dots\cdot(n-k+k)}{1\cdot 2\cdot\dots\cdot k}$$

```
unsigned long long combination(unsigned n, unsigned k) {
    k = min(k, n-k);
    unsigned long long numarator = 1, numitor = 1;
    for (int i = 1; i <= k; i++) {
        numarator = numarator * (n - k + i);
        numitor = numitor * i;
    }
    return numarator / numitor;
}
```

C_n^k — formula directă

$$C_n^k = C_n^{n-k} = \frac{n!}{(n-k)!k!} = \frac{(n-k+1)\cdot(n-k+2)\cdot\dots\cdot(n-k+k)}{1\cdot 2\cdot\dots\cdot k}$$

```
unsigned long long combination(unsigned n, unsigned k) {
    k = min(k, n-k);
    unsigned long long numarator = 1, numitor = 1;
    for (int i = 1; i <= k; i++) {
        numarator = numarator * (n - k + i);
        numitor = numitor * i;
    }
    return numarator / numitor;
}
```

Câte operații se fac?

Observați vreo problemă?

C_n^k — Definiția recursivă

$$C_n^0 = 1, \quad C_n^n = 1, \quad C_n^k = C_{n-1}^k + C_{n-1}^{k-1}$$

```
unsigned long long combination(unsigned n, unsigned k) {
    k = min(k, n-k);
    if (k == 0 || n == k) return 1;
    return combination(n-1, k) + combination(n-1, k-1);
}
```

C_n^k — Definiția recursivă

$$C_n^0 = 1, \quad C_n^n = 1, \quad C_n^k = C_{n-1}^k + C_{n-1}^{k-1}$$

```
unsigned long long combination(unsigned n, unsigned k) {
    k = min(k, n-k);
    if (k == 0 || n == k) return 1;
    return combination(n-1, k) + combination(n-1, k-1);
}
```

Observați vreo problemă?

C_n^k — Definiția recursivă

$$C_n^0 = 1, \quad C_n^n = 1, \quad C_n^k = C_{n-1}^k + C_{n-1}^{k-1}$$

```
unsigned long long combination(unsigned n, unsigned k) {
    k = min(k, n-k);
    if (k == 0 || n == k) return 1;
    return combination(n-1, k) + combination(n-1, k-1);
}
```

Observați vreo problemă?

Câte adunări se fac?

Truc: reținerea valorilor deja calculate

$$C_n^0 = 1, \quad C_n^n = 1, \quad C_n^k = C_{n-1}^k + C_{n-1}^{k-1}$$

```
unsigned long long memo[101][101];

unsigned long long combination(unsigned n, unsigned k) {
    k = min(k, n-k);
    if (k == 0 || n == k) return 1;
    unsigned long long cnk = memo[n][k];
    if (cnk == 0) {
        cnk = combination(n-1, k) + combination(n-1, k-1);
        memo[n][k] = cnk;
    }
    return cnk;
}
```

Truc: reținerea valorilor deja calculate

$$C_n^0 = 1, \quad C_n^n = 1, \quad C_n^k = C_{n-1}^k + C_{n-1}^{k-1}$$

```
unsigned long long memo[101][101];

unsigned long long combination(unsigned n, unsigned k) {
    k = min(k, n-k);
    if (k == 0 || n == k) return 1;
    unsigned long long cnk = memo[n][k];
    if (cnk == 0) {
        cnk = combination(n-1, k) + combination(n-1, k-1);
        memo[n][k] = cnk;
    }
    return cnk;
}
```

Câte adunări se fac?

Truc: tabelare

Triunghiul lui Pascal:

C_0^0					1				
C_1^0	C_1^1				1	1			
C_2^0	C_2^1	C_2^2			1	2	1		
C_3^0	C_3^1	C_3^2	C_3^3		1	3	3	1	
C_4^0	C_4^1	C_4^2	C_4^3	C_4^4	1	4	6	4	1

Truc: tabelare

```
unsigned long long linie_triunghi[101];

unsigned long long combination(unsigned n, unsigned k) {
  k = min(k, n-k);
  if (k == 0 || n == k) return 1;

  linie_triunghi[0] = 1;
  for (int i = 1; i <= n; i++) {
    if (i <= k) linie_triunghi[i] = 1;
    for (unsigned j = min(k, i-1); j > 0; j--)
      linie_triunghi[j] =
          linie_triunghi[j-1] + linie_triunghi[j];
  }
  return linie_triunghi[k];
}
```

Truc: tabelare

```
unsigned long long linie_triunghi[101];

unsigned long long combination(unsigned n, unsigned k) {
  k = min(k, n-k);
  if (k == 0 || n == k) return 1;

  linie_triunghi[0] = 1;
  for (int i = 1; i <= n; i++) {
    if (i <= k) linie_triunghi[i] = 1;
    for (unsigned j = min(k, i-1); j > 0; j--)
      linie_triunghi[j] =
          linie_triunghi[j-1] + linie_triunghi[j];
  }
  return linie_triunghi[k];
}
```

Câte adunări se fac?

Problemă

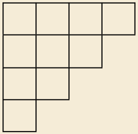
Q2

În câte moduri putem acoperi n trepte cu n dreptunghiuri, unde dreptunghiurile pot avea orice formă?

Acoperirea treptelor cu dreptunghiuri

Exemplu

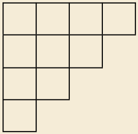
Suprafața în formă de scară cu 4 trepte:

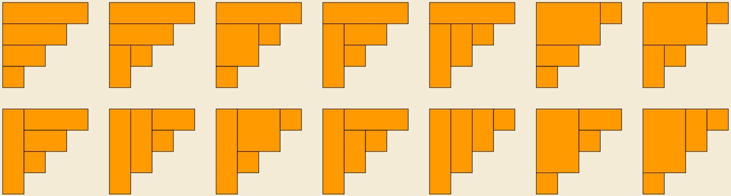


Acoperirea treptelor cu dreptunghiuri

Exemplu

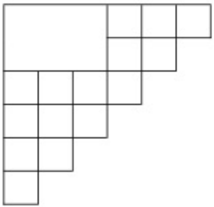
Suprafața în formă de scară cu 4 trepte:





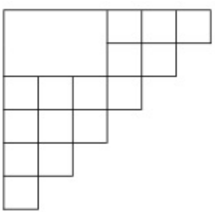
Alegerea triunghiului din colț

Să considerăm cazul cu 6 trepte.



Alegerea triunghiului din colț

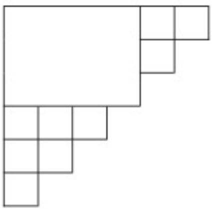
Să considerăm cazul cu 6 trepte.



În acest mod nu se poate obține o acoperire cu 6 dreptunghiuri!

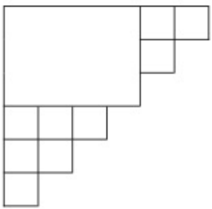
Acoperirea treptelor cu dreptunghiuri

Fiecare dreptunghi trebuie să acopere o treaptă!



Acoperirea treptelor cu dreptunghiuri

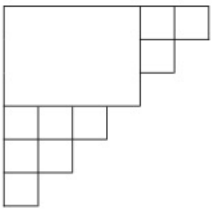
Fiecare dreptunghi trebuie să acopere o treaptă!



Deci, dreptunghiul din colț împarte problema în două.

Acoperirea treptelor cu dreptunghiuri

Fiecare dreptunghi trebuie să acopere o treaptă!



Deci, dreptunghiul din colț împarte problema în două.

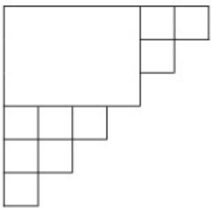
Formula recursivă

Notăm cu C_n numărul de acoperiri pentru n trepte. Se observă relația:

$$C_6 = C_5 + C_1 \cdot C_4 + C_2 \cdot C_3 + C_3 \cdot C_2 + C_4 \cdot C_1 + C_5$$

Acoperirea treptelor cu dreptunghiuri

Fiecare dreptunghi trebuie să acopere o treaptă!



Deci, dreptunghiul din colț împarte problema în două.

Formula recursivă

Notăm cu C_n numărul de acoperiri pentru n trepte. Se observă relația:

$$C_6 = C_5 + C_1 \cdot C_4 + C_2 \cdot C_3 + C_3 \cdot C_2 + C_4 \cdot C_1 + C_5$$

$$C_6 = \sum_{k=1}^{6} C_{k-1} C_{n-k}, \text{ cu } C_0 = 1.$$

Truc: Numerele lui Catalan

Formula recursivă

$$C_n = \sum_{k=1}^{n} C_{k-1} C_{n-k}, \text{ cu } C_0 = 1.$$

- $C_0 = 1$,
- $C_n = C_0 C_{n-1} + C_1 C_{n-2} + \cdots + C_{n-2} C_1 + C_{n-1} C_0$

Truc: Numerele lui Catalan

Formula recursivă

$$C_n = \sum_{k=1}^{n} C_{k-1} C_{n-k}, \text{ cu } C_0 = 1.$$

- $C_0 = 1$,
- $C_n = C_0 C_{n-1} + C_1 C_{n-2} + \cdots + C_{n-2} C_1 + C_{n-1} C_0$

Rezolvarea recursiei

$$C_n = \frac{1}{n+1} C_{2n}^{n}$$

Truc: Numerele lui Catalan

Formula recursivă

$$C_n = \sum_{k=1}^{n} C_{k-1} C_{n-k}, \text{ cu } C_0 = 1.$$

- $C_0 = 1$,
- $C_n = C_0 C_{n-1} + C_1 C_{n-2} + \cdots + C_{n-2} C_1 + C_{n-1} C_0$

Rezolvarea recursiei

$$C_n = \frac{1}{n+1} C_{2n}^{n}$$

Exemplu:

1 1 2 5 14 42 132 429 1430 4862 16796 58786
208012 742900 2674440 9694845 35357670 129644790

Aplicații

Q

În câte moduri putem paranteza produsul $x_0 \cdot x_1 \cdot x_2 \cdot \ldots \cdot x_n$?

Aplicații

Q

În câte moduri putem paranteza produsul $x_0 \cdot x_1 \cdot x_2 \cdot \ldots \cdot x_n$?

Exemplu (n=3)

- $x_0 \cdot (x_1 \cdot (x_2 \cdot x_3))$
- $x_0 \cdot ((x_1 \cdot x_2) \cdot x_3)$
- $(x_0 \cdot x_1) \cdot (x_2 \cdot x_3)$
- $((x_0 \cdot x_1) \cdot x_2) \cdot x_3$
- $(x_0 \cdot (x_1 \cdot x_2)) \cdot x_3$

Aplicații

Q

În câte moduri putem paranteza produsul $x_0 \cdot x_1 \cdot x_2 \cdot \ldots \cdot x_n$?

Rezolvare:

Notăm cu p_n numărul parantezărilor lui $x_0 \cdot x_1 \cdot x_2 \cdot \ldots \cdot x_n$.
Observăm

$$\underset{\substack{\downarrow \\ p_k}}{(x_0 \cdot \ldots \cdot x_k)} \cdot \underset{\substack{\downarrow \\ p_{n-k-1}}}{(x_{k+1} \cdot \ldots \cdot x_n)}$$

Aplicații

Q

În câte moduri putem paranteza produsul $x_0 \cdot x_1 \cdot x_2 \cdot \ldots \cdot x_n$?

Rezolvare:

Notăm cu p_n numărul parantezărilor lui $x_0 \cdot x_1 \cdot x_2 \cdot \ldots \cdot x_n$.

Observăm

$$\underbrace{(x_0 \cdot \ldots \cdot x_k)}_{\downarrow \atop p_k} \cdot \underbrace{(x_{k+1} \cdot \ldots \cdot x_n)}_{\downarrow \atop p_{n-k-1}}$$

Deci

$p_n = \sum_{k=0}^{n-1} p_k \cdot p_{n-k-1} = p_0 \cdot p_{n-1} + p_1 \cdot p_{n-2} + \ldots + p_{n-2} \cdot p_1 + p_{n-1} \cdot p_0$

Aplicații

Q

În câte moduri putem paranteza produsul $x_0 \cdot x_1 \cdot x_2 \cdot \ldots \cdot x_n$?

Rezolvare:

Notăm cu p_n numărul parantezărilor lui $x_0 \cdot x_1 \cdot x_2 \cdot \ldots \cdot x_n$.
Observăm

$$\underbrace{(x_0 \cdot \ldots \cdot x_k)}_{\downarrow \atop p_k} \cdot \underbrace{(x_{k+1} \cdot \ldots \cdot x_n)}_{\downarrow \atop p_{n-k-1}}$$

Deci
$p_n = \sum_{k=0}^{n-1} p_k \cdot p_{n-k-1} = p_0 \cdot p_{n-1} + p_1 \cdot p_{n-2} + \ldots + p_{n-2} \cdot p_1 + p_{n-1} \cdot p_0$

Se observă că $p_n = C_n$, deci numărul parantezărilor unui produs de $n+1$ factori este numărul lui Catalan C_n.

Problemă

Q3

În câte moduri se poate scrie un număr n ca sumă a k numere?

$$n = n_1 + n_2 + \ldots + n_k,$$
$$0 < n_1 \le n_2 \le \ldots \le n_k$$

Truc: Partiția unei mulțimi

O partiție a mulțimii A este o familie de submulțimi $\{A_i\}_{i \in I}$ care verifică următoarele proprietăți:

1. $A_i \neq \emptyset$ pentru orice $i \in I$,
2. $A = \bigcup_{i \in I} A_i$,
3. $A_i \cap A_j = \emptyset$ pentru $i \neq j \in I$.

Mulțimile A_i se numesc clase ale parției.

Truc: Partiția unei mulțimi

Exemplu

Care sunt partițiile mulțimii $A = \{1, 2, 3\}$?

Truc: Partiția unei mulțimi

Exemplu

Care sunt partițiile mulțimii $A = \{1, 2, 3\}$?

- $\{1, 2, 3\}$

Truc: Partiția unei mulțimi

Exemplu

Care sunt partițiile mulțimii $A = \{1, 2, 3\}$?

- $\{1, 2, 3\}$
- $\{1, 2\}$, $\{3\}$
- $\{1, 3\}$, $\{2\}$
- $\{2, 3\}$, $\{1\}$

Truc: Partiția unei mulțimi

Exemplu

Care sunt partițiile mulțimii $A = \{1, 2, 3\}$?

- ☐ $\{1, 2, 3\}$
- ☐ $\{1, 2\}$, $\{3\}$
- ☐ $\{1, 3\}$, $\{2\}$
- ☐ $\{2, 3\}$, $\{1\}$
- ☐ $\{1\}$, $\{2\}$, $\{3\}$

Truc: Numărul partițiilor unei mulțimi

Nr. elem.:	0	1	2	3	4	5	6	7	8	9	10
Nr. partiții:	1	1	2	5	15	52	203	877	4140	21147	115975

Truc: Numărul partițiilor unei mulțimi

Nr. elem.:	0	1	2	3	4	5	6	7	8	9	10
Nr. partiții:	1	1	2	5	15	52	203	877	4140	21147	115975

Numărul lui Bell B_n: numărul partițiilor unei mulțimi cu n elemente

Truc: Numărul partițiilor unei mulțimi

Nr. elem.:	0	1	2	3	4	5	6	7	8	9	10
Nr. partiții:	1	1	2	5	15	52	203	877	4140	21147	115975

Numărul lui Bell B_n: numărul partițiilor unei mulțimi cu n elemente

Numărul lui Stirling de tipul II $S(n, k)$: numărul partițiilor cu k clase ale unei mulțimi cu n elemente

Truc: Numărul partițiilor unei mulțimi

Nr. elem.:	0	1	2	3	4	5	6	7	8	9	10
Nr. partiții:	1	1	2	5	15	52	203	877	4140	21147	115975

Numărul lui Bell B_n: numărul partițiilor unei mulțimi cu n elemente

Numărul lui Stirling de tipul II $S(n,k)$: numărul partițiilor cu k clase ale unei mulțimi cu n elemente

$$B_n = S(n,1) + S(n,2) + \ldots + S(n,n-1) + S(n,n)$$

Truc: Numărul partițiilor unei mulțimi

Exemplu

Care sunt partițiile mulțimii $A = \{1, 2, 3\}$?

- $\{1, 2, 3\}$
- $\{1, 2\}$, $\{3\}$
- $\{1, 3\}$, $\{2\}$
- $\{2, 3\}$, $\{1\}$
- $\{1\}$, $\{2\}$, $\{3\}$

- $S(3, 1) = 1$
- $S(3, 2) = 3$
- $S(3, 3) = 1$

$B_3 = S(3, 1) + S(3, 2) + S(3, 3) = 5$

Truc: Numărul partițiilor unei mulțimi

Exemplu

Care sunt partițiile mulțimii $A = \{1, 2, 3\}$?

- $\{1, 2, 3\}$
- $\{1, 2\}$, $\{3\}$
- $\{1, 3\}$, $\{2\}$
- $\{2, 3\}$, $\{1\}$
- $\{1\}$, $\{2\}$, $\{3\}$

- $S(3, 1) = 1$
- $S(3, 2) = 3$
- $S(3, 3) = 1$

$B_3 = S(3, 1) + S(3, 2) + S(3, 3) = 5$

Observați că $S(n, 1) = 1$, $\quad S(n, n) = 1$

Truc: Numărul partițiilor unei mulțimi

Dacă știm să calculăm $S(n, k)$ pentru orice $k \leq n$

$$\mid * * * \mid * * * \mid \cdots \quad \cdots \mid * * * \mid$$

Truc: Numărul partițiilor unei mulțimi

Dacă știm să calculăm $S(n, k)$ pentru orice $k \leq n$

$$| * * * | * * * | \cdots \quad \cdots | * * * |$$

atunci știm să calculăm și $S(n + 1, k)$:

Truc: Numărul partițiilor unei mulțimi

Dacă știm să calculăm $S(n, k)$ pentru orice $k \leq n$

$$| * * * | * * * | \cdots \quad \cdots | * * * |$$

atunci știm să calculăm și $S(n+1, k)$:

1. $| * * * | * * * | \cdots \quad \cdots | * * * | \bullet |$

Truc: Numărul partițiilor unei mulțimi

Dacă știm să calculăm $S(n, k)$ pentru orice $k \leq n$

$$| * * * | * * * | \cdots \quad \cdots | * * * |$$

atunci știm să calculăm și $S(n+1, k)$:

1. $| * * * | * * * | \cdots \quad \cdots | * * * | \bullet |$
2. $| * * * | * * * | \cdots | * \bullet * | \cdots | * * * |$

Truc: Numărul partițiilor unei mulțimi

Dacă știm să calculăm $S(n, k)$ pentru orice $k \leq n$

$$| * * * | * * * | \cdots \quad \cdots | * * * |$$

atunci știm să calculăm și $S(n+1, k)$:

1. $| * * * | * * * | \cdots \quad \cdots | * * * | \bullet |$
2. $| * * * | * * * | \cdots | * \bullet * | \cdots | * * * |$

Formula recursivă

- $S(n+1, k) = S(n, k-1) + kS(n, k)$
- Condițiile inițiale: $S(1, 1) = 1$ și $S(n, 0) = S(0, k) = 0$

Truc: Numărul partițiilor unei mulțimi în k clase

Input: $n \geq 1$, numărul de elemente al mulțimii M

Output: Un vector S în care $S[k]$ este numărul partițiilor lui M cu k clase

Data: Un vector S de dimensiune n

```
begin
    S[1] := 1
    for i := 2 to n do
        S[i] := 1
        for k := i-1 down to 2 do
            S[k] := k * S[k] + S[k-1]
        end
    end
end
return S
```

Aplicații

Q

Să se determine numărul funcțiilor surjective $f : \{1, \ldots, n\} \to \{1, \ldots, k\}$.

Aplicații

Q

Să se determine numărul funcțiilor surjective
$f : \{1, \dots, n\} \to \{1, \dots, k\}$.

Rezolvare:

Notăm acest număr cu $\sigma(n, k)$. Observăm că

- partiționăm mulțimea $\{1, \dots, n\}$ în k clase în $S(n, k)$ moduri
- ducem câte un element din fiecare partiție în mulțimea $\{1, \dots, k\}$ în $k!$ moduri (permutări de k elemente este $k!$)

Deci răspunsul este $\sigma(n, k) = k!S(n, k)$

Aplicații

Q

Să se determine numărul relațiilor de echivalență care se pot defini pe mulțimea $\{1, \ldots, n\}$.

Aplicații

Q

Să se determine numărul relațiilor de echivalență care se pot defini pe mulțimea $\{1, \ldots, n\}$.

Rezolvare:
Oricărei relații de echivalență R pe $\{1, \ldots, n\}$ îi asociem o partiție a mulțimii $\{1, \ldots, n\}$ astfel:

$$(i, j) \in R \quad \Leftrightarrow \quad i \text{ și } j \text{ sunt în aceeași clasă}$$

Observăm că în acest mod se stabilește o bijecție între mulțimea relațiilor de echivalență și partițiile mulțimii $\{1, \ldots, n\}$.
Atunci răspunsul este dat de numărul partițiilor mulțimii $\{1, \ldots, n\}$.

Aplicații

Q

Să se determine numărul matricilor A pătratice de dimensiune n care conțin doar 0 și 1, și care verifică următoarele proprietăți:

1. $A[i,i] = 1$ or. $1 \leq i \leq n$,
2. $A[i,j] = A[j,i]$ or. $1 \leq i,j \leq n$,
3. $A[i,j] = 1$ și $A[j,k]$=1 implică $A[i,k] = 1$ or. $1 \leq i,j,k \leq n$

Aplicații

Q

Să se determine numărul matricilor A pătratice de dimensiune n care conțin doar 0 și 1, și care verifică următoarele proprietăți:

1. $A[i,i] = 1$ or. $1 \leq i \leq n$,
2. $A[i,j] = A[j,i]$ or. $1 \leq i,j \leq n$,
3. $A[i,j] = 1$ și $A[j,k]$=1 implică $A[i,k] = 1$ or. $1 \leq i,j,k \leq n$

Rezolvare:
Asociem lui A o partiție a mulțimii $\{1,\ldots,n\}$ astfel:

$$A[i,j] = 1 \quad \Leftrightarrow \quad i \text{ și } j \text{ sunt în aceeași clasă}$$

Observăm că în acest mod se stabilește o bijecție între mulțimea matricilor de acest tip și partițiile mulțimii $\{1,\ldots,n\}$.
Atunci răspunsul este dat de numărul partițiilor mulțimii $\{1,\ldots,n\}$.

Q3

În câte moduri se poate scrie un număr n ca sumă a k numere?

$$n = n_1 + n_2 + \ldots + n_k,\ 0 < n_1 \leq n_2 \leq \ldots \leq n_k$$

Q3

În câte moduri se poate scrie un număr n ca sumă a k numere?

$$n = n_1 + n_2 + \ldots + n_k,\ 0 < n_1 \leq n_2 \leq \ldots \leq n_k$$

Descoperirea recursiei:

Dacă $p(n, k)$ este acest număr, observăm că avem două grupuri de partiții:

1. partiții cu 1 în ele, i.e. $1 + ceva$
2. partiții fără 1

Q3

În câte moduri se poate scrie un număr n ca sumă a k numere?

$$n = n_1 + n_2 + \ldots + n_k,\ 0 < n_1 \leq n_2 \leq \ldots \leq n_k$$

Descoperirea recursiei:

Dacă $p(n, k)$ este acest număr, observăm că avem două grupuri de partiții:

1. partiții cu 1 în ele, i.e. $1 + ceva$
2. partiții fără 1

Formula recursivă

$$p(0,0) = 0, \quad p(n,0) = p(0,k) = 0,$$

$$p(n,k) = p(n-1,k-1) + p(n-k,k)$$

Problemă

Q4

În câte moduri putem acoperi o placă de $2 \times N$ cu piese de domino 1×2?

Piese de domino

Exemplu

Pentru o placă de 2×7, o soluție este:

Piese de domino: descoperirea recursiei

Fie F_n numărul acoperirilor unei table $2 \times n$.

Piese de domino: descoperirea recursiei

Fie F_n numărul acoperirilor unei table $2 \times n$.

Cât este F_1?

Piese de domino: descoperirea recursiei

Fie F_n numărul acoperirilor unei table $2 \times n$.

Cât este F_1? $F_1 = 1$

Piese de domino: descoperirea recursiei

Fie F_n numărul acoperirilor unei table $2 \times n$.

Cât este F_1? $F_1 = 1$

Cât este F_2?

Piese de domino: descoperirea recursiei

Fie F_n numărul acoperirilor unei table $2 \times n$.

Cât este F_1? $F_1 = 1$

Cât este F_2? $F_2 = 2$

Piese de domino: descoperirea recursiei

Fie F_n numărul acoperirilor unei table $2 \times n$.

Cât este F_1? $F_1 = 1$

Cât este F_2? $F_2 = 2$

Știind să calculăm F_k pt. or. $k < n$, cât este F_n?

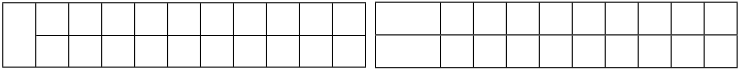
Piese de domino: descoperirea recursiei

Fie F_n numărul acoperirilor unei table $2 \times n$.

Cât este F_1? $F_1 = 1$

Cât este F_2? $F_2 = 2$

Știind să calculăm F_k pt. or. $k < n$, cât este F_n?



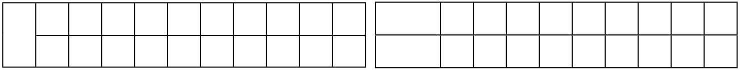
Piese de domino: descoperirea recursiei

Fie F_n numărul acoperirilor unei table $2 \times n$.

Cât este F_1? $F_1 = 1$

Cât este F_2? $F_2 = 2$

Ştiind să calculăm F_k pt. or. $k < n$, cât este F_n?



$$F_n = F_{n-1} + F_{n-2}$$

Truc: Numerele lui Fibonacci

$F_0 = 1$, $F_1 = 1$, $F_n = F_{n-1} + F_{n-2}$ pentru $n \geq 2$

Truc: Numerele lui Fibonacci

$$F_0 = 1,\ F_1 = 1,\ F_n = F_{n-1} + F_{n-2} \text{ pentru } n \geq 2$$

Input: $n \geq 1$
Output: Al n-lea număr din secvența Fibonacci

```
begin
    f1 := 1
    f2 := 1
    for i := 2 to n do
        aux := f1
        f1 := f2
        f2 := aux + f2
    end
end
return f2
```

Aplicații

Reproducerea iepurilor

- Populația începe cu o singură pereche de iepuri.
- Fiecare pereche de iepuri produce o nouă pereche în două luni.
- Presupunem că iepurii nu mor niciodată.
- Câte perechi de iepuri vor fi peste *n* luni?

Aplicații

Reproducerea iepurilor

- Populația începe cu o singură pereche de iepuri.
- Fiecare pereche de iepuri produce o nouă pereche în două luni.
- Presupunem că iepurii nu mor niciodată.
- Câte perechi de iepuri vor fi peste n luni?

Rezolvare:
Dacă f_n este numărul de perechi de iepuri peste n luni, atunci avem relațiile $f_0 = 1$, $f_1 = 1$, $f_{n+1} = f_n + f_{n-1}$ pentru $n \geq 2$.

Aplicații

Q

Care este numărul secvențelor binare de lungime n care nu conțin 0-uri consecutive?

Aplicații

Q

Care este numărul secvențelor binare de lungime n care nu conțin 0-uri consecutive?

Rezolvare:

- Fie s_n numărul secvențelor de lungime n care nu conțin 0-uri consecutive.

Aplicații

Q

Care este numărul secvențelor binare de lungime n care nu conțin 0-uri consecutive?

Rezolvare:

- Fie s_n numărul secvențelor de lungime n care nu conțin 0-uri consecutive.
- Se observă că $s_0 = 1$ (cuvântul vid) și $s_1 = 2$.

Aplicații

Q

Care este numărul secvențelor binare de lungime n care nu conțin 0-uri consecutive?

Rezolvare:

- Fie s_n numărul secvențelor de lungime n care nu conțin 0-uri consecutive.
- Se observă că $s_0 = 1$ (cuvântul vid) și $s_1 = 2$.
- O secvență de lungime n care nu conține 0-uri consecutive se poate termina în:

Aplicații

Q

Care este numărul secvențelor binare de lungime n care nu conțin 0-uri consecutive?

Rezolvare:

- Fie s_n numărul secvențelor de lungime n care nu conțin 0-uri consecutive.
- Se observă că $s_0 = 1$ (cuvântul vid) și $s_1 = 2$.
- O secvență de lungime n care nu conține 0-uri consecutive se poate termina în:
  - 0: $??\ldots?10$, unde $|??\ldots?| = n - 2$

Aplicații

Q

Care este numărul secvențelor binare de lungime n care nu conțin 0-uri consecutive?

Rezolvare:

- Fie s_n numărul secvențelor de lungime n care nu conțin 0-uri consecutive.
- Se observă că $s_0 = 1$ (cuvântul vid) și $s_1 = 2$.
- O secvență de lungime n care nu conține 0-uri consecutive se poate termina în:
  - 0: $??\ldots?10$, unde $|??\ldots?| = n - 2$
  - 1: $??\ldots?1$, unde $|??\ldots?| = n - 1$

Aplicații

Q

Care este numărul secvențelor binare de lungime n care nu conțin 0-uri consecutive?

Rezolvare:

- Fie s_n numărul secvențelor de lungime n care nu conțin 0-uri consecutive.
- Se observă că $s_0 = 1$ (cuvântul vid) și $s_1 = 2$.
- O secvență de lungime n care nu conține 0-uri consecutive se poate termina în:
  - 0: $??\ldots?10$, unde $|??\ldots?| = n-2$
  - 1: $??\ldots?1$, unde $|??\ldots?| = n-1$
- Deci $s_n = s_{n-1} + s_{n-2}$.

Aplicații

Q

Care este numărul submulțimilor lui $\{1, \ldots, n\}$ care nu conțin două numere consecutive?

Aplicații

Q

Care este numărul submulțimilor lui $\{1, \ldots, n\}$ care nu conțin două numere consecutive?

Rezolvare:

- Fie a_n numărul submulțimilor lui $\{1, \ldots, n\}$ care nu conțin două numere consecutive.

Aplicații

Q

Care este numărul submulțimilor lui $\{1, \ldots, n\}$ care nu conțin două numere consecutive?

Rezolvare:

- Fie a_n numărul submulțimilor lui $\{1, \ldots, n\}$ care nu conțin două numere consecutive.
- Se observă că $a_0 = 1$ (mulțimea vidă) și $a_1 = 2$.

Aplicații

Q

Care este numărul submulțimilor lui $\{1, \ldots, n\}$ care nu conțin două numere consecutive?

Rezolvare:

- Fie a_n numărul submulțimilor lui $\{1, \ldots, n\}$ care nu conțin două numere consecutive.
- Se observă că $a_0 = 1$ (mulțimea vidă) și $a_1 = 2$.
- Pentru $n \geq 2$, dacă A este o submulțime a lui $\{1, \ldots, n\}$ care nu conține două numere consecutive, atunci avem două cazuri:

Aplicații

Q

Care este numărul submulțimilor lui $\{1, \ldots, n\}$ care nu conțin două numere consecutive?

Rezolvare:

- Fie a_n numărul submulțimilor lui $\{1, \ldots, n\}$ care nu conțin două numere consecutive.
- Se observă că $a_0 = 1$ (mulțimea vidă) și $a_1 = 2$.
- Pentru $n \geq 2$, dacă A este o submulțime a lui $\{1, \ldots, n\}$ care nu conține două numere consecutive, atunci avem două cazuri:
  - n este în A: Atunci $n - 1$ nu poate să fie în A și $A \setminus \{n\}$ este o submulțime a lui $\{1, \ldots, n-2\}$ care nu conțin două numere consecutive.

Aplicații

Q

Care este numărul submulțimilor lui $\{1, \ldots, n\}$ care nu conțin două numere consecutive?

Rezolvare:

- Fie a_n numărul submulțimilor lui $\{1, \ldots, n\}$ care nu conțin două numere consecutive.
- Se observă că $a_0 = 1$ (mulțimea vidă) și $a_1 = 2$.
- Pentru $n \geq 2$, dacă A este o submulțime a lui $\{1, \ldots, n\}$ care nu conține două numere consecutive, atunci avem două cazuri:
  - *n este în A:* Atunci $n - 1$ nu poate să fie în A și $A \setminus \{n\}$ este o submulțime a lui $\{1, \ldots, n-2\}$ care nu conțin două numere consecutive.
  - *n nu este în A:* Atunci A este o submulțime a lui $\{1, \ldots, n-1\}$ care nu conțin două numere consecutive.

Aplicații

Q

Care este numărul submulțimilor lui $\{1, \ldots, n\}$ care nu conțin două numere consecutive?

Rezolvare:

- Fie a_n numărul submulțimilor lui $\{1, \ldots, n\}$ care nu conțin două numere consecutive.
- Se observă că $a_0 = 1$ (mulțimea vidă) și $a_1 = 2$.
- Pentru $n \geq 2$, dacă A este o submulțime a lui $\{1, \ldots, n\}$ care nu conține două numere consecutive, atunci avem două cazuri:
  - *n este în A*: Atunci $n - 1$ nu poate să fie în A și $A \setminus \{n\}$ este o submulțime a lui $\{1, \ldots, n-2\}$ care nu conțin două numere consecutive.
  - *n nu este în A*: Atunci A este o submulțime a lui $\{1, \ldots, n-1\}$ care nu conțin două numere consecutive.
- Deci $a_n = a_{n-2} + a_{n-1}$.

Problemă

Q6

În câte zerouri se termină 100! ?

$$100! = 1 \cdot 2 \cdot 3 \cdot 4 \ldots \cdot 97 \cdot 98 \cdot 99 \cdot 100$$

În câte zerouri se termină $n!$?

Algoritmul naiv

Input: $n \in \mathbb{N}$

Output: Numărul de zerouri din $n!$

```
begin
    fact := 1
    for i := 2 to n do
        fact := fact * i
    end
    z := 0
    while fact mod 10 = 0 do
        fact := fact div 10
        z := z + 1
    end
end
return z
```

În câte zerouri se termină $n!$?

Algoritmul naiv

Input: $n \in \mathbb{N}$

Output: Numărul de zerouri din $n!$

```
begin
    fact := 1
    for i := 2 to n do
        fact := fact * i
    end
    z := 0
    while fact mod 10 = 0 do
        fact := fact div 10
        z := z + 1
    end
end
return z
```

Observați vreo problemă?

În câte zerouri se termină $n!$?

- Zerourile de la sfârșitul lui $n!$ apar din produse de 2 și 5

În câte zerouri se termină $n!$?

- Zerourile de la sfârșitul lui $n!$ apar din produse de 2 și 5
- Descompunem $n!$ în factori primi: $n! = 2^{k_2} * 3^{k_3} * 5^{k_5} * \cdots$

În câte zerouri se termină $n!$?

- Zerourile de la sfârșitul lui $n!$ apar din produse de 2 și 5
- Descompunem $n!$ în factori primi: $n! = 2^{k_2} * 3^{k_3} * 5^{k_5} * \cdots$
- Numărul zerourile de la sfârșitul lui $n!$ este $min(k_2, k_5)$.

În câte zerouri se termină $n!$?

- Zerourile de la sfârșitul lui $n!$ apar din produse de 2 și 5
- Descompunem $n!$ în factori primi: $n! = 2^{k_2} * 3^{k_3} * 5^{k_5} * \cdots$
- Numărul zerourile de la sfârșitul lui $n!$ este $min(k_2, k_5)$.
- Sunt mai puțini factori 5 decât 2, deci nr. de zerouri căutat este k_5.

În câte zerouri se termină $n!$?

- Zerourile de la sfârșitul lui $n!$ apar din produse de 2 și 5
- Descompunem $n!$ în factori primi: $n! = 2^{k_2} * 3^{k_3} * 5^{k_5} * \cdots$
- Numărul zerourile de la sfârșitul lui $n!$ este $min(k_2, k_5)$.
- Sunt mai puțini factori 5 decât 2, deci nr. de zerouri căutat este k_5.
- $k_5 =$ nr. numerelor mai mici ca n divizibile cu 5 $+$
  nr. numerelor mai mici ca n divizibile cu 25 $+$
  nr. numerelor mai mici ca n divizibile cu 125 $+$ ...

$$k_5 = \sum_{i \geq 1} \left\lfloor \frac{n}{5^i} \right\rfloor, \qquad \text{unde} \qquad \left\lfloor \frac{n}{5^{i+1}} \right\rfloor = \left\lfloor \frac{\left\lfloor \frac{n}{5^i} \right\rfloor}{5} \right\rfloor$$

În câte zerouri se termină $n!$?

$$k_5 = \sum_{i \geq 1} \left\lfloor \frac{n}{5^i} \right\rfloor, \qquad \text{unde} \quad \left\lfloor \frac{n}{5^{i+1}} \right\rfloor = \left\lfloor \frac{\left\lfloor \frac{n}{5^i} \right\rfloor}{5} \right\rfloor$$

Input: $n \in \mathbb{N}$

Output: Numărul de zerouri din $n!$

```
begin
    z := 0
    while n ≠ 0 do
        n := n div 5
        z := z + n
    end
end
return z
```

Problemă

Q7

Pentru două numere $n \geq 1$ și $p \geq 1$, $n \geq p$, găsiți x și y cu proprietatea

$$xn \; + \; yp = cmmdc(n, p)$$

Truc: Algoritmul lui Euclid

Algoritmul lui Euclid (scăderi repetate)

Input: $n, p \in \mathbb{N}, n \geq p$

Output: $cmmdc(n, p)$

```
begin
    while n ≠ p do
        if n ≥ p then
            n := n - p
        else
            p := p - n
        end
    end
end
return n
```

Truc: Algoritmul lui Euclid

Algoritmul lui Euclid (ultimul rest nenul)

Input: $n, p \in \mathbb{N}, n \geq p$

Output: $cmmdc(n, p)$

```
begin
    while p ≠ 0 do
        aux := p
        p := n mod p
        n := aux
    end
end
return n
```

Truc: Algoritmul lui Euclid

Algoritmul începe cu n și p și constă în calcularea

- unei secvențe $q_1, \ldots, q_k$ de câturi și a
- unei secvențe $r_0, \ldots, r_{k+1}$ de resturi

astfel încât:

$r_0 = n$

$r_1 = p$

$\ldots$

$r_{i+1} = r_{i-1} - q_i r_i$ și $0 \leq r_{i+1} < r_i$

$\ldots$

Algoritmul se termină când un rest r_{k+1} este 0 și cmmdc-ul este r_k.

Câturile $q_1, \ldots, q_k$ nu sunt folosite!

Truc: Algoritmul lui Euclid

Input: $n, p \in \mathbb{N}, n \geq p$

Output: $cmmdc(n, p)$

```
begin
    r := n
    r' := p
    while r' ≠ 0 do
        ct := r div r'
        aux := r
        r := r'
        r' := aux - ct * r'
    end
end
return r
```

Truc: Algoritmul lui Euclid extins

Algoritmul lui Euclid extins permite calcularea a doi întregi x și y astfel încât

$$xn + yp = cmmdc(n, p) \qquad \text{(Identitatea lui Bézout)}$$

Truc: Algoritmul lui Euclid extins

Algoritmul începe cu n și p, și constă în adăugarea a două secvențe suplimentare

- $x_0, \ldots, x_k$ și
- $y_0, \ldots, y_k$ d

astfel încât

$$r_i = x_i * n + y_i * p$$

$r_0 = n \quad r_1 = p \quad \ldots$
$x_0 = 1 \quad x_1 = 0 \quad \ldots$
$y_0 = 0 \quad y_1 = 1 \quad \ldots$

Formula recursivă

$r_{i+1} = r_{i-1} - q_i r_i$ astfel încâ t $0 \leq r_{i+1} < r_i$
$x_{i+1} = x_{i-1} - q_i x_i$
$y_{i+1} = y_{i-1} - q_i y_i$

Truc: Algoritmul lui Euclid extins

Algoritmul se termină când un rest r_{k+1} este 0.
În acest caz obținem:

- r_k este cmmdc(n,p)
- Coeficienții Bézout sunt x_k și y_k, adică:

$$x_k n + y_k p = cmmdc(n, p) = r_k$$

Algoritmul lui Euclid extins

Input: n, p
Output: r=$cmmdc(n,p)$ și x, y, astfel încât $x * a + y * b = r$
begin

```
    x := 1   x' := 0
    y := 0   y' := 1
    r := n   r' := p
    while r' ≠ 0 do
        ct := r div r'
        aux := r   r := r'   r' := aux - ct * r'
        aux := x   x := x'   x' := aux - ct * x'
        aux := y   y := y'   y' := aux - ct * y'
    end
```

end
return r,x,y

Aplicații

Q

Să se verifice dacă n este inversabil modulo p.
În caz afirmativ, să se determine $n^{-1}(mod\ p)$.

Aplicații

Q

Să se verifice dacă n este inversabil modulo p.
În caz afirmativ, să se determine $n^{-1}(mod\ p)$.

Rezolvare:

- Algoritmul lui Euclid permite calcularea a doi întregi x și y astfel încât $xn + yp = cmmdc(n, p)$.
- Dacă $cmmdc(n, p) = 1$, atunci $n^{-1}(mod\ p) = x$.

Problemă

Q8

Care este cel mai mic număr natural n care împărțit la 2 dă restul 1, împărțit la 3 dă restul 2, împărțit la 5 dă restul 3, iar împărțit la 7 dă restul 4?

Problemă

Q8

Care este cel mai mic număr natural n care împărțit la 2 dă restul 1, împărțit la 3 dă restul 2, împărțit la 5 dă restul 3, iar împărțit la 7 dă restul 4?

Q

Fie $p_1, \ldots, p_k$ numere naturale prime între ele și
$0 \leq s_i < p_i$, oricare $1 \leq i \leq k$.

Determinați $s \geq 0$ astfel încât $s = s_i \ (mod\ p_i)$ oricare $1 \leq i \leq k$.

Truc: Teorema chineză a resturilor

Teoremă

$$s = \sum_{i=1}^{k} (s_i \cdot P_i \cdot x_i),$$

unde $P_i = \frac{p_1 \cdots p_k}{p_i}$ *și* $x_i = P_i^{-1} (mod\ p_i)$ *oricare* $1 \leq i \leq k$.

Truc: Teorema chineză a resturilor

Teoremă

$$s = \sum_{i=1}^{k} (s_i \cdot P_i \cdot x_i),$$

unde $P_i = \frac{p_1 \cdots p_k}{p_i}$ *și* $x_i = P_i^{-1} (mod\ p_i)$ *oricare* $1 \leq i \leq k$.

Observați că x_i poate fi calculat folosind Algoritmul lui Euclid extins.

Truc: Teorema chineză a resturilor

$s = \sum_{i=1}^{k}(s_i P_i x_i)$, unde $P_i = \frac{p_1 \cdots p_k}{p_i}$ și $x_i = P_i^{-1}(mod\, p_i)$ or. i.

Exemplu

$p_1 = 2,\ p_2 = 3,\ p_3 = 5,\ p_4 = 7$
$s_1 = 1,\ s_2 = 2,\ s_3 = 3,\ s_4 = 4$

Truc: Teorema chineză a resturilor

$s = \sum_{i=1}^{k}(s_i P_i x_i)$, unde $P_i = \frac{p_1 \cdots p_k}{p_i}$ și $x_i = P_i^{-1}(mod\, p_i)$ or. i.

Exemplu

$p_1 = 2,\ p_2 = 3,\ p_3 = 5,\ p_4 = 7$
$s_1 = 1,\ s_2 = 2,\ s_3 = 3,\ s_4 = 4$

$P = 2 * 3 * 5 * 7 = 210$

Truc: Teorema chineză a resturilor

$s = \sum_{i=1}^{k}(s_i P_i x_i)$, unde $P_i = \frac{p_1 \cdots p_k}{p_i}$ și $x_i = P_i^{-1}(mod\, p_i)$ or. i.

Exemplu

$p_1 = 2,\ p_2 = 3,\ p_3 = 5,\ p_4 = 7$
$s_1 = 1,\ s_2 = 2,\ s_3 = 3,\ s_4 = 4$

$P = 2 * 3 * 5 * 7 = 210$

$P_1 = 105,\ 1 = 105^{-1}\ (mod\, 2),\ x_1 = 1$

Truc: Teorema chineză a resturilor

$s = \sum_{i=1}^{k}(s_i P_i x_i)$, unde $P_i = \frac{p_1 \cdots p_k}{p_i}$ și $x_i = P_i^{-1}(mod\, p_i)$ or. i.

Exemplu

$p_1 = 2,\ p_2 = 3,\ p_3 = 5,\ p_4 = 7$
$s_1 = 1,\ s_2 = 2,\ s_3 = 3,\ s_4 = 4$

$P = 2 * 3 * 5 * 7 = 210$

$P_1 = 105,\ 1 = 105^{-1}\ (mod\, 2),\ x_1 = 1$

$P_2 = 70,\ 1 = 70^{-1}\ (mod\, 3),\ x_2 = 1$

Truc: Teorema chineză a resturilor

$s = \sum_{i=1}^{k}(s_i P_i x_i)$, unde $P_i = \frac{p_1 \cdots p_k}{p_i}$ și $x_i = P_i^{-1}(mod\, p_i)$ or. i.

Exemplu

$p_1 = 2,\ p_2 = 3,\ p_3 = 5,\ p_4 = 7$
$s_1 = 1,\ s_2 = 2,\ s_3 = 3,\ s_4 = 4$

$P = 2 * 3 * 5 * 7 = 210$

$P_1 = 105,\ 1 = 105^{-1}\ (mod\, 2),\ x_1 = 1$

$P_2 = 70,\ 1 = 70^{-1}\ (mod\, 3),\ x_2 = 1$

$P_3 = 42,\ 3 = 42^{-1}\ (mod\, 5),\ x_3 = 3$

Truc: Teorema chineză a resturilor

$s = \sum_{i=1}^{k}(s_i P_i x_i)$, unde $P_i = \frac{p_1 \cdots p_k}{p_i}$ și $x_i = P_i^{-1}(mod\, p_i)$ or. i.

Exemplu

$p_1 = 2,\ p_2 = 3,\ p_3 = 5,\ p_4 = 7$
$s_1 = 1,\ s_2 = 2,\ s_3 = 3,\ s_4 = 4$

$P = 2 * 3 * 5 * 7 = 210$

$P_1 = 105,\ 1 = 105^{-1}\ (mod\, 2),\ x_1 = 1$

$P_2 = 70,\ 1 = 70^{-1}\ (mod\, 3),\ x_2 = 1$

$P_3 = 42,\ 3 = 42^{-1}\ (mod\, 5),\ x_3 = 3$

$P_4 = 30,\ 4 = 30^{-1}\ (mod\, 7),\ x_4 = 4$

Truc: Teorema chineză a resturilor

$s = \sum_{i=1}^{k}(s_i P_i x_i)$, unde $P_i = \frac{p_1 \cdots p_k}{p_i}$ și $x_i = P_i^{-1}(mod\, p_i)$ or. i.

Exemplu

$p_1 = 2,\ p_2 = 3,\ p_3 = 5,\ p_4 = 7$
$s_1 = 1,\ s_2 = 2,\ s_3 = 3,\ s_4 = 4$

$P = 2 * 3 * 5 * 7 = 210$

$P_1 = 105,\ 1 = 105^{-1}\ (mod\, 2),\ x_1 = 1$

$P_2 = 70,\ 1 = 70^{-1}\ (mod\, 3),\ x_2 = 1$

$P_3 = 42,\ 3 = 42^{-1}\ (mod\, 5),\ x_3 = 3$

$P_4 = 30,\ 4 = 30^{-1}\ (mod\, 7),\ x_4 = 4$

Deci $s = 1 * 105 * 1 + 2 * 70 * 1 + 3 * 42 * 3 + 4 * 30 * 4 = 1109$

Baftă la examen!